

Utah State University

DigitalCommons@USU

All Graduate Theses and Dissertations, Fall
2023 to Present

Graduate Studies

12-2023

Development, Implementation, and Optimization of a Modern, Subsonic/Supersonic Panel Method

Cory D. Goates

Utah State University, cory.goates@usu.edu

Follow this and additional works at: <https://digitalcommons.usu.edu/etd2023>



Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Goates, Cory D., "Development, Implementation, and Optimization of a Modern, Subsonic/Supersonic Panel Method" (2023). *All Graduate Theses and Dissertations, Fall 2023 to Present*. 30.

<https://digitalcommons.usu.edu/etd2023/30>

This Dissertation is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations, Fall 2023 to Present by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.



DEVELOPMENT, IMPLEMENTATION, AND OPTIMIZATION OF A
MODERN, SUBSONIC/SUPERSONIC PANEL METHOD

by

Cory D. Goates

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Mechanical Engineering

Approved:

Douglas F. Hunsaker, Ph.D.
Major Professor

Barton L. Smith, Ph.D.
Committee Member

Matthew W. Harris, Ph.D.
Committee Member

Som Dutta, Ph.D.
Committee Member

Todd K. Moon, Ph.D.
Committee Member

D. Richard Cutler, Ph.D.
Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2023

Copyright © Cory D. Goates 2023

All Rights Reserved

ABSTRACT

Development, Implementation, and Optimization of a
Modern, Subsonic/Supersonic Panel Method

by

Cory D. Goates, Doctor of Philosophy

Utah State University, 2023

Major Professor: Douglas F. Hunsaker, Ph.D.
Department: Mechanical and Aerospace Engineering

Modern high-speed computing has made viable the application of automatic design and optimization schemes to supersonic vehicle design. To ensure optimal results can be obtained in a timely manner, fast and accurate aerodynamic simulation methods are needed. Panel methods present a significant opportunity for filling this need more effectively than most aerodynamic models now in use. Unlike higher-fidelity methods, such as finite-volume Navier-Stokes solvers, panel methods require only a discretization of the configuration surface, making meshing straightforward and leading to run times on the order of less than a minute on standard desktop computers. In addition, numerous investigations have shown that panel methods produce accurate results for many supersonic flows of interest. However, supersonic panel methods have seen little improvement over the past few decades and so are being largely superseded by more modern and user-friendly, albeit computationally expensive, Navier-Stokes and Euler methods. To address the shortcomings of extant supersonic panel methods, this work addresses the theoretical development and implementation of a modern supersonic panel method. The implementation of a panel method on unstructured meshes is discussed. Varying boundary condition formulations and orders of singularity distributions are derived, implemented, and compared for accuracy, robustness, and speed.

In addition, optimizations to a panel method, such as quickly solving the linear system of equations and efficiently determining domains of dependence, are implemented and examined. A significant amount of work has also been done to unify and clarify the available literature regarding supersonic panel methods. Results from the pilot code are presented and compared against analytic and experimental data to demonstrate the accuracy of the implementation described here. Overall, the supersonic panel method is shown to be fast, numerically robust, and accurate for many flows of interest.

(408 pages)

PUBLIC ABSTRACT

Development, Implementation, and Optimization of a
Modern, Subsonic/Supersonic Panel Method

Cory D. Goates

In the early stages of aircraft design, engineers consider many different design concepts, examining the trade-offs between different component arrangements and sizes, thrust and power requirements, etc. Because so many different designs are considered, it is best in the early stages of design to use simulation tools that are fast; accuracy is secondary. A common simulation tool for early design and analysis is the panel method. Panel methods were first developed in the 1950s and 1960s with the advent of modern computers. Despite being reasonably accurate and very fast, their development was abandoned in the late 1980s in favor of more complex and accurate simulation methods. The panel methods developed in the 1980s are still in use by aircraft designers today because of their accuracy and speed. However, they are cumbersome to use and limited in applicability. The purpose of this work is to reexamine panel methods in a modern context. In particular, this work focuses on the application of panel methods to supersonic aircraft (a supersonic aircraft is one that flies faster than the speed of sound). Various aspects of the panel method, including the distributions of the unknown flow variables on the surface of the aircraft and efficiently solving for these unknowns, are discussed. Trade-offs between alternative formulations are examined and recommendations given. This work also serves to bring together, clarify, and condense much of the the literature previously published regarding panel methods so as to assist future developers of panel methods.

To the glory of God.

*"...if ye cannot understand them it will be because ye ask not, neither do ye knock;
wherefore, ye are not brought into the light..." - 2 Nephi 32:4*

ACKNOWLEDGMENTS

The greatest appreciation is due to my wife, Kathryn. She married me knowing that I wanted to pursue a PhD and has encouraged and helped me every step of the way. Thank you for all the sacrifices and patience. When the going got rough, she helped push me through. I don't know that I'll ever be able to express to her just how grateful I am for everything she has done and continues to do for me. Thanks also to my son, Cory Simon, for inspiring me to work hard and for always wanting to come sit by me while I work.

My advisor, Dr. Doug Hunsaker, has been a miracle in my life. He has patiently mentored and supported me throughout both my undergraduate and graduate work. Thank you for your constant confidence in me while also giving me the criticism and feedback I needed.

Thanks to my parents for setting high expectations and loving me with their whole hearts. I am proud to be the second "Dr. Goates" in the family. My siblings have also been a great inspiration, encouragement, and support throughout my education. Special thanks to Josh for introducing me to the AeroLab and mentoring me, and to Jeremy for helping me debug my code and run much of the data presented in this work.

Throughout my education, I have been blessed to work with many excellent professors. Thank you Dr. Geordie Richards and Dr. Todd Moon for introducing me to the beauty of mathematical depth and rigor. Thank you Dr. Randy Christensen for believing in me. Thank you Dr. Matt Harris for asking the hard questions. Thank you Dr. Bart Smith and Dr. Areti Kiara for sharing your fascination with and passion for engineering. Thank you Dr. Thom Fronk for the heart and soul you put into teaching all of us. Thank you Dr. Som Dutta for your enthusiasm for my work and encouraging me to explore.

Thanks go to Dr. Christian Bolander (PhD USU 2022) for laying a significant portion of the groundwork for this research. Much of the material presented in the Introduction was produced in collaboration with him, and we had a great time doing so. Many thanks also to Justin Schoppe of Texas A&M University for his help in refining and optimizing the

recursive domain of dependence algorithm. Thanks also to Ted Giblette (MS USU 2019) for providing the original inspiration for this work and mentoring me as a engineering software developer. Thanks also to all the other AeroLab students (Nathan Hoch, Sydney Perkins, Josh Hurwitz, Spencer Adams, and Conner Carlile) who will be carrying this work forward and who have inspired and enabled me to produce a polished final result to my research.

Many thanks also to Vivek Ahuja and Roy Hartfield of Research in Flight for partnering with us on this research. They were instrumental in securing much of the funding for this research and provided tremendous feedback and encouragement during every step of the work.

Special thanks go to Joan Rudd, our Graduate Program Coordinator for Mechanical and Aerospace Engineering. Her undying patience and positive attitude have been a tremendous blessing. Thanks for all the scheduled meetings, endless answered questions, and fun barbeques.

Thanks be to God for giving me the capability and talents to do this work. I pray my actions will bless His children.

Cory D. Goates

CONTENTS

	Page
ABSTRACT	iii
PUBLIC ABSTRACT	v
ACKNOWLEDGMENTS	vii
LIST OF TABLES	xiv
LIST OF FIGURES	xvi
NOTATIONS	xxvii
1 INTRODUCTION	1
1.1 Prandtl-Glauert Equation	1
1.1.1 Essential Theorems	2
1.1.2 Bernoulli's Equation	7
1.1.3 Speed of Sound	8
1.1.4 Bernoulli's Equation for Steady Flows	9
1.1.5 Full (Nonlinear) Potential Equation of Motion	10
1.1.6 Mach Number	13
1.1.7 Linearization to Obtain the Prandtl-Glauert Equation	14
1.1.8 Linearized Flow Properties	20
1.2 Characteristic Surfaces	22
1.2.1 Characteristic Surfaces of the Full Potential Equation	22
1.2.2 Characteristic Surfaces of the Prandtl-Glauert Equation	25
1.2.3 Sub- and Superinclined Surfaces	26
1.3 Method of Green's Functions	28
1.3.1 General Solution for Subsonic Flows	30
1.3.2 General Solution for Supersonic Flows	33
1.3.3 Unified Boundary Integral Equation	35
1.4 Panel Methods	37
1.4.1 Solving the Boundary Integral Equation	37
1.4.2 Differences Between Subsonic and Supersonic Flows	39
1.4.3 History	40
1.5 Existing Difficulties and Weaknesses	43
1.5.1 Lack of clarity regarding the need for higher-order singularity distributions	43
1.5.2 Lack of clarity regarding trade-offs between boundary-condition formulations	45
1.5.3 Use of structured meshes	46
1.5.4 Unreliable treatment of superinclined panels	47

1.5.5	Minimal use of efficient matrix algorithms	47
1.5.6	Appearance of expansive shocks	48
1.5.7	Constant characteristic angles	48
1.6	Alternative Aerodynamic Methods	48
1.6.1	Field-Panel Methods	49
1.6.2	Full Potential Methods	50
1.6.3	Euler and Navier-Stokes Methods	51
1.7	Research Objectives	51
1.7.1	Objective 1: Characterize the impact of varying singularity distributions on the accuracy, speed, and robustness of an unstructured panel method.	52
1.7.2	Objective 2: Characterize the impact of boundary condition formulation on the accuracy, speed, and robustness of an unstructured panel method.	52
1.7.3	Objective 3: Assess the performance of various matrix solvers for the unique matrix equation appearing in supersonic panel methods. . . .	53
1.7.4	Objective 4: Identify a viable path forward for the development of a recursive domain of dependence search algorithm and assess its impact on solution speed.	53
1.8	Pilot Code	54
1.9	Current Scope	54
2	BOUNDARY CONDITIONS	55
2.1	Dirichlet Boundary Condition Formulations	55
2.1.1	Morino Formulation	58
2.1.2	Source-Free Formulation for Zero Normal Mass Flux	60
2.1.3	Solving for the Unknown Doublet Strengths	61
2.2	Superinclined Panels	63
2.2.1	Difficulties with Superinclined Panels	63
2.2.2	Boundary Conditions on Superinclined Panels	64
2.2.3	Calculation of Velocity Influence Coefficients	66
2.3	Alternative Formulations	67
2.3.1	Neumann Formulations	67
2.3.2	Source-Free Formulation for Zero Normal Velocity	68
3	CONFIGURATION MODELING	70
3.1	Unstructured Meshes	70
3.2	Coordinate Systems	71
3.2.1	Transformation from Global to Compressibility Coordinates	72
3.2.2	Transformation from Compressibility to Scaled Coordinates	74
3.2.3	Transformation from Global to Local-Scaled Coordinates	74
3.3	Panel Geometric Parameters	76
3.4	Edge Search and Characterization	76
3.5	Wake Modeling	78
3.5.1	Automatic Supersonic Wake Truncation	80
3.6	Mesh Mirroring	81
3.7	Control Point Placement	82

3.7.1	Control Points on Wake-Shedding Edges	84
3.7.2	Control Point Placement Correction	87
4	DOMAIN OF DEPENDENCE CALCULATIONS	92
4.1	Point Domain of Dependence Formulas	92
4.2	Panels and Domains of Dependence	93
4.3	Efficient Domain of Dependence Calculation	94
5	SOURCE AND DOUBLET PANELS	101
5.1	Types of Distributions and their Characteristics	102
5.2	Influence Calculations	104
5.2.1	Influence Integrals	105
5.2.2	Fundamental Integrals	107
5.2.3	Influences in Terms of Fundamental Integrals	108
5.2.4	Geometric Quantities	112
5.2.5	Calculation of H Integrals	117
5.2.6	Calculation of F Integrals	123
5.3	Strength and Parameter Spaces	127
5.4	Transformations for Lower-Order Distributions	129
5.4.1	Linear Doublet Distribution	129
5.4.2	Constant Source Distribution	130
5.5	Transformations for Higher-Order Distributions	131
5.5.1	Quadratic Doublet Distribution	131
5.5.2	Linear Source Distribution	145
5.6	Wake Modeling with Higher-Order Distributions	151
6	SOLVING THE LINEAR SYSTEM OF EQUATIONS	153
6.1	Structure of the Supersonic AIC Matrix	154
6.2	System Solvers	161
6.2.1	LU Decomposition	161
6.2.2	Block-Iterative Solvers	162
6.2.3	Generalized Minimum Residual Solver	163
6.2.4	Novel QR Decomposition Solvers	163
6.3	Solver Comparison	169
6.3.1	Cone	171
6.3.2	Double-Wedge Wing	175
6.3.3	Wing-Body-Nacelle Combination	178
6.3.4	Time Complexities	185
6.3.5	Higher-Order Distributions and Lower Bandwidth	186
6.4	Discussion	188
7	POST-PROCESSING	190
7.1	Surface Velocity Calculation	190
7.2	Surface Pressure Calculation	191
7.2.1	Vacuum Pressure Coefficient	192
7.2.2	Stagnation Pressure Coefficient	193
7.2.3	Subsonic Pressure Corrections	193

7.3	Force and Moment Integration	195
7.3.1	Force Integration	195
7.3.2	Moment Integration	198
7.4	Accuracy Determination Based on Pressure Rules	201
8	NUMERICAL BEHAVIORS AND SENSITIVITIES	203
8.1	Sensitivity to Control Point Offset	204
8.1.1	Incompressible Sphere	204
8.1.2	Supersonic Double-Wedge Wing	205
8.1.3	Supersonic Spindle	208
8.1.4	Discussion	209
8.2	Order of Grid Convergence	210
8.2.1	Incompressible Sphere	211
8.2.2	Supersonic Spindle	213
8.2.3	Supersonic Double-Wedge Wing	215
8.2.4	Supersonic Delta Wing	218
8.2.5	Discussion	223
8.3	Sensitivity to Panel Regularity	224
8.3.1	Incompressible Sphere	224
8.3.2	Supersonic Spindle	226
8.3.3	Discussion	230
9	MODELING STUDIES	231
9.1	Sphere in Compressible Subsonic Flow	231
9.2	ONERA M6 Wing at High-Subsonic Mach Numbers	234
9.3	Diamond (Double-Wedge) Airfoil	242
9.4	Supersonic Cone Flow	250
9.5	Delta Wing	262
9.5.1	Comparison with PAN AIR	279
9.6	Flow-Through Nacelles	289
9.6.1	Subsonic Nacelle	289
9.6.2	Supersonic Nacelle	293
9.7	Discussion	300
9.7.1	Singularity Distributions	301
9.7.2	Boundary Condition Formulations	303
10	CONCLUSION	304
10.1	Summary of Work	304
10.2	Outcome of Objectives	307
10.3	Recommendations for Users of MachLine	309
10.4	Novel Contributions	311
10.5	Limitations	312
10.6	Future Work	313
10.6.1	Improved Implementation of Higher-Order Singularity Distributions	313
10.6.2	Improvements to the FQRUP Solver	315
10.6.3	Superinclined Panels	316
10.6.4	Improved Kutta Condition	316

10.6.5	Higher-Order Wake Panels	317
10.6.6	Wake Relaxation	317
10.6.7	Wake-Fuselage Stitching	318
10.6.8	Adaptive p -Refinement	319
10.6.9	Neumann Boundary Conditions	319
10.6.10	Multipole Methods for Influence Calculations	320
REFERENCES		321
APPENDICES		332
A	PROOF THAT THE INVERSE OF R_β OR R_B SATISFIES THE PRANDTL-GLAUERT EQUATION	333
B	ALTERNATE DERIVATION OF THE UNIFIED BOUNDARY INTEGRAL EQUATION	336
C	PILOT CODE	339
C.1	Code Description	339
C.2	Running the Code	341
C.3	Studies	348
D	PSEUDOCODE FOR THE DOMAIN OF DEPENDENCE ALGORITHMS	354
E	INFLUENCE COEFFICIENT CALCULATION DEVELOPMENT METHODOLOGY	356
F	DERIVATION OF A USEFUL INTEGRAL IDENTITY	360
G	PROOF OF SUBSONIC $F(1, 1, 1)$ FORMULA	362
H	H AND F INTEGRAL RECURSION RELATIONS	364
H.1	Definitions	364
H.2	H Recursion Relations	365
H.3	F Recursion Relations	367
I	CALCULATION OF C INTEGRALS	371
CURRICULUM VITAE		375

LIST OF TABLES

Table		Page
2.1	Source strengths and desired inner potentials for the two boundary condition formulations.	61
6.1	Formulas for generating a fast Givens rotation, taken from [1]. Note, as the final step in each case, A_{ij} must be set to zero.	167
6.2	Freestream Mach numbers and system dimension (N) for the three configurations considered.	170
6.3	Lower bandwidths for the sorted AIC matrix as a function of configuration, system dimension, and the type of singularity distribution used.	187
8.1	Labels used for the different combinations of boundary condition formulation and singularity distribution used in this work.	203
8.2	Orders of convergence for the sphere in incompressible flow. The uncertainty band represents one standard deviation.	213
8.3	Orders of convergence for the spindle in supersonic flow. The uncertainty band represents the standard deviation over all Mach numbers.	215
8.4	Orders of convergence for the double-wedge wing in supersonic flow with a maximum continuity angle of 1° . The uncertainty band is the standard deviation across all Mach numbers and angles of attack.	216
8.5	Orders of convergence for the double-wedge wing in supersonic flow with a maximum continuity angle of 45° . The uncertainty band is the standard deviation across all Mach numbers and angles of attack.	216
8.6	Orders of convergence for the delta wing in supersonic flow. The uncertainty band the standard deviation across all angles of attack.	222
9.1	Total computation run times in seconds averaged across freestream conditions for the sphere in compressible flow.	234
9.2	Total computation run times in seconds averaged across freestream conditions for the ONERA M6 wing.	241
9.3	Total run times in seconds averaged across all Mach numbers and angles of attack for the double-wedge wing.	250

9.4	Total run times in seconds averaged across all Mach numbers for the 2.5° half-angle cone.	262
9.5	Total run times in seconds averaged across all Mach numbers for the 5° half-angle cone.	262
9.6	Total run times in seconds averaged across all Mach numbers for the 10° half-angle cone.	262
9.7	Total run times in seconds averaged across all Mach numbers for the 15° half-angle cone.	262
C.1	Options for the flow section of the input file.	345
C.2	Options for the geometry section of the input file.	350
C.3	Options for the solver section of the input file.	351
C.4	Options for the post-processing section of the input file.	352
C.5	Options for the output section of the input file.	353

LIST OF FIGURES

Figure	Page
1.1 Examples of (a) sub- and (b) superinclined surfaces showing the intersection of the domain of dependence with the surface.	27
1.2 Region of integration for the supersonic Green's theorem.	34
3.1 Rotation from the global to the compressible coordinate system.	73
3.2 Side views of different edge cases showing when a wake will and will not be shed.	78
3.3 Wake panels shed from the trailing edge of a swept wing, illustrating the basic wake-generation algorithm in MachLine.	79
3.4 The three possible cases for mesh mirroring in MachLine: a) Case 1: no mirroring, b) Case 2: mirrored mesh and symmetric flow condition where the mirrored half of the mesh does not need to be explicitly modeled, and c) Case 3: mirrored mesh and asymmetric flow condition where the mirrored half of the mesh is explicitly modeled and has different flow results from the original half.	82
3.5 The two contiguous groups of panels (light gray and dark gray) touching the vertex highlighted in blue on either side of a string of wake-shedding edges (red).	86
3.6 Placement of control points for vertices belonging to wake-shedding edges. .	87
3.7 (a) Convex and (b) non-convex sets of panels. For the convex set of panels, the panels (dark gray) are all on the boundary of their convex closure (the boundaries of the convex closure not belonging to panels are shaded light gray). This is not the case with the non-convex set as the panels outlined by the dashed lines are not on the boundary of the convex closure.	88
4.1 Mach wedge region of a supersonic edge. The Mach wedge is the region in which a point may be influenced by the edge but falls outside the Mach cones for either endpoint.	94
4.2 A case where a superinclined panel belongs to the domain of dependence of a control point when none of its vertices or edges do.	94

4.3	Diagram of the domains of dependence for two points showing how the domain of dependence of one point is entirely encompassed by that of the other point.	95
4.4	Comparison of the orders of complexity for the brute-force and recursive domain of dependence searches for three different meshes as a function of freestream Mach number.	97
5.1	Points used to define the quadratic doublet distribution over a panel in the six-point-fit method. The six mesh vertices marked in red are the points used for the gray panel.	133
5.2	Points used to defined the quadratic doublet distribution over a panel with one discontinuous edge in the six-point-fit method. The points used are highlighted in red. The left edge is discontinuous.	135
5.3	Mesh paneling for which the six-point-fit method fails. The gray panel is the one under consideration, and the five red vertices are those used to determine the panel's doublet strength distribution. The top edge in this figure is considered discontinuous.	139
5.4	Points used (red) to determine midpoint strengths (blue) for a) continuous and b) discontinuous edges.	141
6.1	Diagram of the general structure of an AIC matrix.	154
6.2	Heatmaps showing the magnitudes of the AIC matrix coefficients for a) a standard subsonic case and b) a high-aspect-ratio, 5° half-angle, double-wedge wing in supersonic flow. The scale is logarithmic. Black regions signify AIC elements which are identically zero.	155
6.3	Heatmap showing the magnitudes of the AIC matrix coefficients for a 10° half-angle cone in supersonic flow ($M_\infty = 1.5$). The scale is logarithmic. Black regions signify AIC elements which are identically zero.	156
6.4	Heatmap representing the AIC matrix coefficients for the wing represented in Fig. 6.2b after sorting the mesh vertices using Algorithm 1.	160
6.5	Vertices on an unstructured mesh that must be considered in the second sorting pass for determining the sorting parameter for the blue vertex. Lower-order distributions require considering the red vertices. Higher-order distributions require considering the red and yellow vertices. If the edge shared by both panels touching a yellow vertex and the blue vertex is discontinuous, then that yellow vertex is not considered.	162
6.6	Medium mesh for the 10° half-angle cone.	170

6.7	Medium mesh for the double-wedge wing.	170
6.8	Medium mesh for the wing-body-nacelle combination.	170
6.9	Average run times for the cone case with the system sorted.	172
6.10	Average run times for the cone case with the system unsorted.	173
6.11	Average residual norms for the cone case with the system sorted. Dashed line indicates the iterative solver termination tolerance.	173
6.12	Average residual norms for the cone case with the system unsorted. Dashed line indicates the iterative solver termination tolerance.	174
6.13	Solution residual versus run time for all solvers considered for the coarse cone mesh.	175
6.14	Solution residual versus run time for all solvers considered for the medium cone mesh.	175
6.15	Solution residual versus run time for all solvers considered for the fine cone mesh.	176
6.16	Average run times for the wing case with the system sorted.	177
6.17	Average run times for the wing case with the system unsorted.	177
6.18	Average residual norms for the wing case with the system sorted. Dashed line indicates the iterative solver termination tolerance.	178
6.19	Average residual norms for the wing case with the system unsorted. Dashed line indicates the iterative solver termination tolerance.	179
6.20	Solution residual versus run time for all solvers considered for the coarse wing mesh.	179
6.21	Solution residual versus run time for all solvers considered for the medium wing mesh.	180
6.22	Solution residual versus run time for all solvers considered for the fine wing mesh.	180
6.23	Heatmap representing the AIC matrix for the medium wing-body-nacelle configuration mesh at $M_\infty = 2$	181
6.24	Average run times for the wing-body-nacelle case with the system sorted.	181
6.25	Average run times for the wing-body-nacelle case with the system unsorted.	182

6.26	Average residual norms for the wing-body-nacelle case with the system sorted. Dashed line indicates the iterative solver termination tolerance.	183
6.27	Average residual norms for the wing-body-nacelle case with the system unsorted. Dashed line indicates the iterative solver termination tolerance.	184
6.28	Solution residual versus run time for all solvers considered for the coarse wing-body-nacelle mesh.	184
6.29	Solution residual versus run time for all solvers considered for the medium wing-body-nacelle mesh.	185
6.30	Solution residual versus run time for all solvers considered for the fine wing-body-nacelle mesh.	185
6.31	Solver time complexities averaged across the three configurations with and without diagonal preconditioning. The uncertainty bands shown represent the standard deviation between the three configurations.	187
8.1	Norm of force coefficient vector as a function of control point offset (specified as a ratio of the average edge lengths) for a sphere with (a) a mesh with approximately 7200 panels and (b) a mesh with approximately 1700 panels.	205
8.2	Norm of force coefficient vector as a function of control point offset (specified directly) for a sphere with (a) a mesh with approximately 7200 panels and (b) a mesh with approximately 1700 panels.	206
8.3	(a) Axial and (b) off-axis force coefficient magnitudes on a double-wedge wing as a function of k_1 (specified as a ratio) for all boundary condition formulations and singularity distributions.	207
8.4	(a) Axial and (b) off-axis force coefficient magnitudes on a finely-meshed (2200 panels) double-wedge wing as a function of k_1 (specified directly).	207
8.5	(a) Axial and (b) off-axis force coefficient magnitudes on a coarsely-meshed (800 panels) double-wedge wing as a function of k_1 , with k_1 specified as the offset directly, rather than as a ratio.	208
8.6	(a) Axial and (b) off-axis force coefficient magnitudes on a finely-meshed (12500 panels) supersonic spindle as a function of k_1	209
8.7	(a) Axial and (b) off-axis force coefficient magnitudes on a coarsely-meshed (722 panels) supersonic spindle as a function of k_1	210
8.8	Example sphere mesh showing the radial symmetry about the x -axis.	212

8.9	Norm of the force coefficient acting on the sphere as a function of the average panel characteristic length for each singularity distribution and boundary condition formulation.	212
8.10	Fractional error relative to the most-refined mesh in the axial force coefficient on the spindle as a function of mesh average characteristic length at various Mach numbers for each of the boundary condition formulations and singularity distributions.	214
8.11	Example mesh for the double-wedge wing.	215
8.12	Isentropic pressure distribution at the tip of the double-wedge wing at $M_\infty = 1.5$ and $\alpha = 0^\circ$. The freestream flow is from left to right. Color scale ranges from dark red (high pressure) to dark blue (low pressure).	218
8.13	Fractional error relative to the most-refined mesh in the axial force coefficient on the double-wedge wing as a function of mesh average characteristic length at various Mach numbers and angles of attack for each of the boundary condition formulations and singularity distributions.	219
8.14	Fractional error relative to the most-refined mesh in the normal force coefficient on the double-wedge wing as a function of mesh average characteristic length at various Mach numbers and angles of attack for each of the boundary condition formulations and singularity distributions.	220
8.15	Computational meshes used for the delta wing grid convergence study. . . .	221
8.16	Grid convergences rates for the supersonic delta wing over a range of angles of attack.	221
8.17	Pressure coefficient predictions for a coarse, randomly-paneled sphere in incompressible flow compared to the analytic result.	226
8.18	Pressure coefficient predictions for a fine, randomly-paneled sphere in incompressible flow compared to the analytic result.	227
8.19	Example mesh for the randomly-paneled spindle with 800 vertices.	228
8.20	Pressure coefficient predictions for a randomly-paneled spindle in supersonic flow compared to results from the method of characteristics [2].	229
9.1	Surface velocity predictions over the front half of a sphere at $M_\infty = 0.4$ compared to the second-order result of Tamada [3].	232
9.2	Surface velocity predictions over the front half of a sphere at $M_\infty = 0.5$ compared to the second-order result of Tamada [3].	233

9.3	Results at 20% semispan on the ONERA M6 wing at $M_\infty = 0.7$ and $\alpha = 0.06^\circ$ from MachLine using both the direct subsonic calculation and the Prandtl-Glauert correction. Experimental results are included from [4].	236
9.4	Top-down view of the isentropic pressure distribution over the ONERA M6 wing at $M_\infty = 0.7$ and $\alpha = 0.06^\circ$ using varying mesh resolutions. Color scale ranges from $C_P \approx 1.1$ (dark red) to $C_P \approx -2.9$ (dark blue).	237
9.5	View of the leading edges of the ONERA M6 wing at 0° angle of attack, facing downstream, at the medium mesh resolution with higher-order singularity distributions. Color scale ranges from approximately $C_P \approx -1.6$ (blue) to $C_P \approx 0.1$ (red).	238
9.6	Results at 90% semispan on the ONERA M6 wing at $M_\infty = 0.7$ and $\alpha = 5^\circ$ from MachLine using both the direct subsonic calculation and the Prandtl-Glauert correction. Experimental results are included from [4].	239
9.7	Results at 99% semispan on the ONERA M6 wing at $M_\infty = 0.7$ and $\alpha = 5^\circ$ from MachLine using both the direct subsonic calculation and the Prandtl-Glauert correction. Experimental results are included from [4].	241
9.8	Results at 90% semispan on the ONERA M6 wing at $M_\infty = 0.84$ and $\alpha = 0^\circ$ from MachLine using both the direct subsonic calculation and the Prandtl-Glauert correction. Experimental results are included from [4].	242
9.9	Airfoil section used for the diamond wing with surfaces labeled.	243
9.10	Surface pressures for a 5° half-angle double-wedge airfoil over angle of attack at $M_\infty = 1.5$ predicted by MachLine (discrete points) and shock-expansion theory (lines). The line style indicates which surface the pressure coefficient is over (see Fig. 9.9). Solid: lower-fore. Dashed: upper-fore. Dash-dot: lower-aft. Dotted: upper-aft.	245
9.11	Surface pressures for a 5° half-angle double-wedge airfoil over angle of attack at $M_\infty = 2$ predicted by MachLine (discrete points) and shock-expansion theory (lines). The line style indicates which surface the pressure coefficient is over (see Fig. 9.9). Solid: lower-fore. Dashed: upper-fore. Dash-dot: lower-aft. Dotted: upper-aft.	246
9.12	Surface pressures for a 5° half-angle double-wedge airfoil over angle of attack at $M_\infty = 3$ predicted by MachLine (discrete points) and shock-expansion theory (lines). The line style indicates which surface the pressure coefficient is over (see Fig. 9.9). Solid: lower-fore. Dashed: upper-fore. Dash-dot: lower-aft. Dotted: upper-aft.	247

9.13	Surface pressures for a 5° half-angle double-wedge airfoil over angle of attack at $M_\infty = 5$ predicted by MachLine (discrete points) and shock-expansion theory (lines). The line style indicates which surface the pressure coefficient is over (see Fig. 9.9). Solid: lower-fore. Dashed: upper-fore. Dash-dot: lower-aft. Dotted: upper-aft.	248
9.14	Tip region on the 15° half-angle cone at $M_\infty = 2$ showing the isentropic pressure distribution calculated on the fine mesh when using the Morino formulation and higher-order singularity distributions.	251
9.15	Tip region on the 15° half-angle cone at $M_\infty = 2$ showing the isentropic pressure distribution calculated on the fine mesh when using the source-free formulation and higher-order singularity distributions.	252
9.16	Rendering of the isentropic pressure distribution at the tip of the 10° cone at $M_\infty = 1.7$ using Morino formulation and higher-order distributions on the fine mesh. The erroneous panel is the blue one (extremely low pressure) at the bottom of the second row from the tip.	253
9.17	First and second rows of panels at the tip of the cone showing both the original (light gray) and mirrored (dark gray) halves of the mesh.	254
9.18	Average surface pressure coefficients for supersonic flow over a 2.5° half-angle cone as a function of Mach number compared to results from the Taylor-MacColl equation [5]. Error bars represent the standard deviation in the pressure coefficient over the whole surface of the cone.	255
9.19	Aft portion of the 2.5° half-angle cone at $M_\infty = 3.3$ showing erroneous pressure results at the trailing edge as a function of mesh density. The color scale ranges from $C_P \approx 0.002$ (dark blue) to $C_P \approx 0.7$ (dark red).	256
9.20	Isentropic pressure distribution over the 2.5° half-angle cone at $M_\infty = 1.7$ using higher-order singularity distributions and the source-free formulation on the finest mesh.	257
9.21	Average surface pressure coefficients for supersonic flow over a 5° half-angle cone as a function of Mach number compared to results from the Taylor-MacColl equation [5]. Error bars represent the standard deviation in the pressure coefficient over the whole surface of the cone.	258
9.22	Average surface pressure coefficients for supersonic flow over a 10° half-angle cone as a function of Mach number compared to results from the Taylor-MacColl equation [5]. Error bars represent the standard deviation in the pressure coefficient over the whole surface of the cone.	259

9.23	Average surface pressure coefficients for supersonic flow over a 15° half-angle cone as a function of Mach number compared to results from the Taylor-MacColl equation [5]. Error bars represent the standard deviation in the pressure coefficient over the whole surface of the cone.	260
9.24	Top-down view of delta wing mesh.	263
9.25	Erroneous isentropic pressure distribution over the top of the Love delta wing at 0° angle of attack using the Morino formulation and higher-order singularity distributions with the heavily cosine-clustered mesh. The color scale ranges from $C_P \approx 1.84$ (dark red) to $C_P \approx -0.54$ (dark blue). The erroneous pressure results were due to extremely fine panels near the leading edge and line of maximum thickness.	264
9.26	Grid convergence of the normal force coefficient (C_y) for Love's delta wing [6] over a range of angles of attack and $M_\infty = 1.62$	266
9.27	Grid convergence of the axial force coefficient (C_x) for Love's delta wing [6] over a range of angles of attack and $M_\infty = 1.62$	267
9.28	Lift coefficient for Love's delta wing [6] over a range of angles of attack and $M_\infty = 1.62$ from MachLine compared to experimental data reported by Love [6] and predictions obtained from CPanel, as reported by Davis [7]. . .	268
9.29	Drag coefficient for Love's delta wing [6] over a range of angles of attack and $M_\infty = 1.62$ from MachLine compared to experimental data reported by Love [6] and predictions obtained from CPanel, as reported by Davis [7]. . .	269
9.30	Isentropic pressure distribution at 22.5% semispan and 0° angle of attack from MachLine compared to experimental data reported by Love [6] and an analytic solution based on a linearized theory presented by Hayes [8]. . . .	270
9.31	Isentropic pressure distribution at 64.1% semispan and 0° angle of attack from MachLine compared to experimental data reported by Love [6] and an analytic solution based on a linearized theory presented by Hayes [8]. . . .	271
9.32	Isentropic pressure distribution over the top of the Love delta wing at 0° angle of attack using the Morino formulation and higher-order singularity distributions showing the characteristic lines (shocks) emanating from the bow and the point of maximum thickness. The color scale indicates high pressures with dark red and low pressures with dark blue.	272
9.33	Isentropic pressure distribution at 22.5% semispan and 4.1° angle of attack from MachLine compared to experimental data reported by Love [6]. . . .	273
9.34	Isentropic pressure distribution at 64.1% semispan and 4.1° angle of attack from MachLine compared to experimental data reported by Love [6]. . . .	274

9.35	Isentropic pressure distribution at 22.5% semispan and 10.75° angle of attack from MachLine compared to experimental data reported by Love [6].	275
9.36	Isentropic pressure distribution at 64.1% semispan and 10.75° angle of attack from MachLine compared to experimental data reported by Love [6].	276
9.37	Relative error in C_x on the delta wing for the different boundary conditions and singularity distributions at four different angles of attack as a function of computation time.	277
9.38	Relative error in C_y on the delta wing for the different boundary conditions and singularity distributions at four different angles of attack as a function of computation time.	278
9.39	Top view of the coarsest delta wing mesh used in MachLine to compare against PAN AIR. For use in MachLine, each quadrilateral panel has been split into two triangles. In both MachLine and PAN AIR, the mesh was mirrored across the root plane.	280
9.40	Pressure distributions over the delta wing using the medium mesh at 0° angle of attack using lower-order singularity distributions showing the differences due to which diagonal was used to subdivide each quadrilateral panel for MachLine. The color scale indicates high pressures with dark red and low pressures with dark blue.	281
9.41	Pressure distributions over the delta wing using the medium mesh at 0° angle of attack using higher-order singularity distributions showing the differences due to which diagonal was used to subdivide each quadrilateral panel for MachLine. The color scale indicates high pressures with dark red and low pressures with dark blue.	281
9.42	Orders of grid convergence using PAN AIR and the various formulations in MachLine for the delta wing as a function of angle of attack.	282
9.43	Pressure results over the outer portion of the delta wing at 0° angle of attack using the higher-order singularity distributions and the Morino formulation showing the erroneous pressure predictions at the wingtip. The color scale ranges from $C_P \approx 1.84$ (dark red) to $C_P \approx -0.54$ (dark blue).	283
9.44	Total (a) axial and (b) normal force coefficients acting on the delta wing (fine mesh) as a function of angle of attack predicted by PAN AIR and MachLine.	283
9.45	Total (a) axial and (b) normal force coefficients acting on the delta wing (medium mesh) as a function of angle of attack predicted by PAN AIR and MachLine.	284

9.46	Surface isentropic pressure coefficients at 6% semispan on the delta wing at varying angles of attack from MachLine and PAN AIR.	286
9.47	Surface isentropic pressure coefficients at 50% semispan on the delta wing at varying angles of attack from MachLine and PAN AIR.	287
9.48	Surface isentropic pressure coefficients at 73% semispan on the delta wing at varying angles of attack from MachLine and PAN AIR.	288
9.49	Average run times for MachLine and PAN AIR as a function of mesh resolution for the supersonic delta wing.	289
9.50	Meshes showing effect of varying mesh refinement in the circumferential direction going from (a) coarse to (c) fine.	290
9.51	Meshes showing effect of varying mesh refinement in the axial direction going from (a) coarse to (c) fine.	290
9.52	Surface isentropic pressure coefficients along an axial slice of the NACA 0010 nacelle in incompressible flow for varying levels of axial mesh refinement. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).	291
9.53	Surface isentropic pressure coefficients along an axial slice of the NACA 0010 nacelle in incompressible flow for varying levels of circumferential mesh refinement. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).	292
9.54	Surface isentropic pressure coefficients along an axial slice of the NACA 0010 nacelle in incompressible flow for varying levels of mesh refinement in both the circumferential and axial directions. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black). . . .	294
9.55	Surface isentropic pressure coefficients along an axial slice of the inside of the supersonic nacelle at $M_\infty = 1.5$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).	295
9.56	Surface isentropic pressure coefficients along an axial slice of the outside of the supersonic nacelle at $M_\infty = 1.5$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).	297
9.57	Surface isentropic pressure coefficients along an axial slice of the inside of the supersonic nacelle at $M_\infty = 2$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).	298
9.58	Surface isentropic pressure coefficients along an axial slice of the outside of the supersonic nacelle at $M_\infty = 2$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).	299

- 9.59 Surface isentropic pressure coefficients along an axial slice of the inside of the supersonic nacelle at $M_\infty = 2.5$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black). 300
- 9.60 Surface isentropic pressure coefficients along an axial slice of the outside of the supersonic nacelle at $M_\infty = 2.5$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black). 301

NOTATIONS

Variables

a	edge perpendicular in-plane distance
$[A_{c \rightarrow s}]$	transformation matrix from the compressibility to the scaled frame
$[A_{g \rightarrow c}]$	transformation matrix from the global to the compressibility frame
$[A_{g \rightarrow ls}]$	transformation matrix from the global to the local-scaled frame
$[AIC]$	aerodynamic influence coefficient matrix
b	panel edge inclination indicator
B	supersonic compressibility scaling factor
$[B]$	compressible scaling matrix
B_l	matrix lower bandwidth
c	speed of sound
$\hat{\mathbf{c}}$	freestream direction vector (points downstream)
C_F	force coefficient vector
C_{ij}	panel moment integrals for calculating force and moment coefficients
C_M	moment coefficient vector
C_P	pressure coefficient
c_s	sonic speed
c_v	constant volume specific heat
$[C]$	compressible scaling matrix
D_P	domain of dependence of point P
$\frac{\partial}{\partial \bar{n}}$	conormal derivative
$\frac{D}{Dt}$	material derivative
e	specific internal energy
$F_i(M, N, K)$	fundamental edge integral for edge i
g	edge perpendicular distance

G	Green's function
h	specific enthalpy (Chapter 1) or height above panel in ls coordinates (Chapter 5)
$H(M, N, K)$	fundamental panel integral
J	area scaling factor going from global to local-scaled coordinates
k_1	control point offset
K	circulation
l_1, l_2	edge tangential distances
l_{avg}	average length of adjacent edges
l_{ref}	reference length for calculating moment coefficients
M	Mach number
$\{M\}$	panel doublet strength vector
\mathbf{n}	surface or panel normal vector
$\hat{\mathbf{n}}$	panel edge outward normal vector
$\tilde{\mathbf{n}}$	conormal vector
p	static pressure
$\mathbf{P} = (x, y, z)$	evaluation point
\mathbf{P}_c	control point location
\mathbf{P}_{cent}	panel centroid
Q	heat added
$\mathbf{Q} = (\xi, \eta, \zeta)$	integration point
r	panel inclination indicator
\mathbf{r}	position vector
R	subsonic/supersonic compressible radius
R_β	subsonic compressible radius
R_B	supersonic compressible (hyperbolic) radius
s	flow type indicator
S	surface bounding the flow region of interest

$\{S\}$	panel source strength vector
t	time
T	temperature
$[T_\mu]$	doublet strength-parameter transformation matrix
$[T_\sigma]$	source strength-parameter transformation matrix
$\hat{\mathbf{t}}$	panel edge tangential vector
\mathbf{u}	total velocity
\mathbf{u}'	perturbation velocity
U	freestream velocity magnitude
\mathbf{U}	freestream velocity
V	flow region of interest or fluid volume
\mathbf{w}	reduced current velocity
α	angle of attack
β	subsonic compressibility scaling factor
γ	ratio of specific heats
ζ	vorticity
η	specific entropy
κ	flow-dependent integration factor
μ	doublet strength
$\{\mu\}$	panel doublet distribution vector
ρ	density
σ	source strength
$\{\sigma\}$	panel source distribution vector
ϕ	perturbation potential
ϕ_d	doublet-induced perturbation potential
ϕ_i	inner perturbation potential
ϕ_∞	freestream potential
ϕ_s	source-induced perturbation potential

Φ	total velocity potential
Φ_s	total velocity potential on outside of S
Subscripts	
∞	freestream property
$*$	stagnation property
0	strength at panel centroid
ξ	derivative in panel local-scaled ξ coordinate
η	derivative in panel local-scaled η coordinate
Superscripts	
c	compressibility coordinate system
g	global (mesh) coordinate system
ls	local-scaled coordinate system
s	scaled coordinate system

CHAPTER 1

INTRODUCTION

The name “panel method” comes from these treatments of the body shape. We at Douglas did not invent this rather appropriate name... - A. M. O. Smith [9]

The primary objective of this research effort is to identify the best approach to modeling supersonic flows using a panel method. Panel methods are a fast way to approximately model inviscid flow fields about complex configurations. As such, they are particularly well-suited to preliminary design and optimization of aircraft (see for example [10–19]).

Panel methods are based on a linear model of fluid behavior. Rather than the Navier-Stokes or Euler equations, panel methods are based on Laplace’s equation (for incompressible flows) or the Prandtl-Glauert equation (for compressible flows). To find solutions to these equations, the method of Green’s functions is employed.

1.1 Prandtl-Glauert Equation

The Prandtl-Glauert equation was developed independently by Ludwig Prandtl and Hermann Glauert. Glauert presented a two-dimensional version of this equation in 1928 [20]. Glauert used this equation to analyze the effect of compressibility on airfoil sections at the tips of aircraft propellers, which even on low-speed aircraft, approached high-subsonic Mach numbers.

A thorough derivation of the Prandtl-Glauert equation in three dimensions is given by Ward in his book *Linearized Theory of Steady High-Speed Flow* [21]. The derivation presented here is taken primarily from Ward. To arrive at the Prandtl-Glauert equation, the fluid under consideration is first assumed to be a continuous medium. The particles of the fluid are then given a specified particle velocity \mathbf{u} , and every point of the fluid is also given a density, ρ , pressure, p , specific energy, e , specific enthalpy, h , specific entropy, η ,

and temperature, T . Generally, these fluid properties will be expressed in spatial coordinates. However, some instances require use of material coordinates. Any use of material coordinates will be explicitly stated. Throughout this section, Einstein summation notation is used to simplify some of the steps of the derivation (for an introduction to Einstein summation notation, see Chapter 2 in [22]).

The most basic equations of fluid motion used to derive the Prandtl-Glauert equation are the continuity equation and Euler's equations of motion, given by [21, 23]

$$\frac{\partial \rho}{\partial t} + (\rho u_i)_{,i} = 0 \quad (1.1)$$

$$\frac{\partial u_i}{\partial t} + u_j u_{i,j} + \frac{1}{\rho} p_{,i} = 0 \quad (1.2)$$

respectively. In using Euler's equations of motion, it is assumed that the fluid is inviscid. It is also assumed that the fluid is a perfect gas with constant specific heats, which has the equation of state [21]

$$p = \rho^\gamma e^{\frac{\eta}{c_v}} \quad (1.3)$$

where γ is the ratio of specific heats and c_v is the specific heat of the gas at constant volume.

1.1.1 Essential Theorems

A handful of theorems need to be derived as precursors to obtaining the Prandtl-Glauert equation.

Euler's Momentum Theorem

Adding Eq. (1.1) multiplied by \mathbf{u} to Eq. (1.2) multiplied by ρ yields

$$u_i \frac{\partial \rho}{\partial t} + u_i (\rho u_j)_{,j} + \rho \frac{\partial u_i}{\partial t} + \rho u_j u_{i,j} + \rho \frac{1}{\rho} p_{,i} = 0 \quad (1.4)$$

Using the product rule, the second and fourth terms, and the first and third terms in this equation may be combined. Doing so yields

$$\frac{\partial \rho u_i}{\partial t} + (\rho u_i u_j)_{,j} + p_{,i} = 0 \quad (1.5)$$

Equation (1.5) is then integrated over the volume of the fluid, V , bounded by a closed surface S , which is fixed relative to the reference frame. Doing so yields

$$\int_V \frac{\partial \rho u_i}{\partial t} dV + \int_V [(\rho u_i u_j)_{,j} + p_{,i}] dV = 0 \quad (1.6)$$

Using Gauss' divergence theorem, the second integral becomes

$$\int_V \frac{\partial \rho u_i}{\partial t} dV + \int_S (\rho u_i u_j + p) n_i dS = 0 \quad (1.7)$$

where S is the boundary of V and n_i is the outward normal vector to S . Equation (1.7) is called Euler's momentum theorem [21] and states that, for an inviscid fluid, the time rate of change of momentum within V is opposite the momentum flux across and the pressure acting on the surface S .

Crocco's Theorem

The thermodynamic state of a fluid is completely specified if only two of the state variables are known. All other state variables may then be found from these two. Here, the specific entropy, η , and the density, ρ , are specified as the independent state variables. Thus, the pressure may be written as $p = f(\eta, \rho)$. An important property of all real fluids is that [21]

$$\left(\frac{dp}{d\rho}\right)_\eta = \frac{\partial f(\eta, \rho)}{\partial \rho} > 0 \quad (1.8)$$

That is, for a constant entropy process, an increase in pressure in the fluid also yields an increase in density and vice versa.

If a small amount of heat, δQ , is added to the fluid, the specific internal energy, e will increase and the fluid will also expand according to

$$\delta Q = de + pd(1/\rho) \quad (1.9)$$

If this heat is supplied reversibly then it will also hold that

$$\delta Q = Td\eta \quad (1.10)$$

Combining these two relations gives

$$de = Td\eta - pd(1/\rho) \quad (1.11)$$

If the flow is adiabatic (no heat added) and inviscid (no heat generated by viscosity or transferred by conduction), then $\delta Q = 0$. In this case, $d\eta = 0$ for any given particle throughout its motion. This can be expressed using the material derivative as

$$\frac{D\eta}{Dt} = 0 \quad (1.12)$$

A flow satisfying these criteria is said to be *isentropic*. An isentropic flow is one where the entropy of each fluid particle does not change along its path through the flow field.

Recall that the specific enthalpy, h is defined as $h = e + p/\rho$ [24]. Using Eq. (1.11), the differential enthalpy may be written in terms of the independent state variables (entropy η and density ρ)

$$dh = de + pd(1/\rho) + (1/\rho)dp = Td\eta - pd(1/\rho) + pd(1/\rho) + (1/\rho)dp$$

$$dh = Td\eta + (1/\rho)dp \quad (1.13)$$

For the flow under consideration, it is assumed that all particles have the same equation of state and so Eq. (1.13) holds for all particles. Therefore, Eq. (1.13) holds true for any neighboring particles in a continuous flow and over infinitesimally small distances in any direction throughout a continuous flow. Thus, Eq. (1.13) may be written using the gradient operator, as in

$$\nabla h = T\nabla\eta + \frac{1}{\rho}\nabla p \quad (1.14)$$

Equation (1.2) may also be rewritten using the independent thermodynamic variables. First, it is recognized that $u_j u_{i,j} = \frac{1}{2}(u_j u_j)_{,i} - (u_j u_{j,i} - u_j u_{i,j})$. Using this result in Euler's momentum theorem (Eq. (1.5)) results in

$$\begin{aligned} \frac{\partial u_i}{\partial t} + u_j u_{j,i} - u_j u_{j,i} + u_j u_{i,j} + \frac{1}{\rho} p_{,i} &= 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \frac{1}{2} \nabla(\mathbf{u} \cdot \mathbf{u}) - \mathbf{u} \times (\nabla \times \mathbf{u}) + \frac{1}{\rho} \nabla p &= 0 \end{aligned} \quad (1.15)$$

Using Eq. (1.14) in Eq. (1.15), this becomes

$$\frac{\partial u_i}{\partial t} + \left(\frac{1}{2} u_j u_j + h \right)_{,i} - u_j (u_{j,i} - u_{i,j}) = T \eta_{,i} \quad (1.16)$$

Equation (1.16) is known as Crocco's theorem [21]. It expresses a relation between the vorticity of a fluid, defined as $\boldsymbol{\zeta} = \nabla \times \mathbf{u} = u_{j,i} - u_{i,j}$, and its change in entropy.

Kelvin's Circulation Theorem

Potential methods, such as panel methods, are applied to irrotational flows. To understand irrotationality, the concept of circulation must first be understood. By definition, the circulation, K , around a closed curve, C , lying entirely in the fluid is [21]

$$K = \oint_C u_i dx_i \quad (1.17)$$

If C moves with the fluid (i.e. fixed in material coordinates), then the time rate of change of K is

$$\frac{DK}{Dt} = \frac{D}{Dt} \oint_C u_i dx_i = \oint_C \frac{Du_i}{Dt} dx_i \quad (1.18)$$

Using the material form of Euler's equation of motion [21], this becomes

$$\frac{DK}{Dt} = \oint_C -(1/\rho)p_{,i} dx_i = \oint_C (T\eta_{,i} - h_{,i}) dx_i \quad (1.19)$$

When integrated over the closed curve C , the term $-h_{,i}$ in the integrand vanishes because h is a single-valued function. For a homentropic flow (that is, a flow where the entropy is constant everywhere), $\eta_{,i} = 0$, and the time rate of change of circulation is

$$\frac{DK}{Dt} = 0 \quad (1.20)$$

This result is called Kelvin's circulation theorem [21] and states that, for a homentropic flow, the circulation around any closed curve is constant for all time.

Using Stokes' theorem, the circulation may also be expressed as

$$K = \oint_C u_i dx_i = \int_S \zeta_i n_i dS \quad (1.21)$$

where S is any closed surface having C as its boundary, \mathbf{n} is the normal to S , and $\boldsymbol{\zeta}$ is the fluid vorticity (defined previously). Thus, circulation and vorticity are related.

1.1.2 Bernoulli's Equation

Most flows of interest in aeronautics problems originate as a uniform (homotropic) flow, and so it is assumed that this is the case here. This is in addition to the previous assumptions that the flow is also adiabatic and inviscid, meaning it is isentropic. Note that even though the flow is adiabatic, it will not necessarily be isothermal as the temperature will vary with pressure according to the ideal gas relations. Since the particles began in a homotropic flow, the entropy of all particles will always be the same and the flow is homotropic for all time.

A uniform initial flow also means that the vorticity of the flow is initially zero everywhere, since $u_{,i} \equiv 0$ in a uniform flow. Thus, the circulation about any curve C in the flow is also zero. By Kelvin's circulation theorem (Eq. ((1.20))), the circulation about any curve C convected with the flow will remain zero for all time. From Eq. (1.21), it must then hold that the vorticity of the flow is zero everywhere and for all time (i.e. $\zeta \equiv 0$). Thus, for any inviscid, adiabatic flow which originates in a uniform flow, the flow is homotropic and irrotational everywhere and for all time.

When the flow is irrotational, the velocity can be expressed as the gradient of a scalar potential function, as in

$$\mathbf{u} = \nabla\Phi \quad (1.22)$$

where Φ is the velocity potential. Imposing irrotationality on a homotropic flow, the right-hand side and the third term on the left-hand side of Crocco's theorem (Eq. (1.16)) vanish, leaving

$$\left(\frac{\partial\Phi}{\partial t} + h + \frac{1}{2}u_i u_i \right)_{,i} = 0 \quad (1.23)$$

This expression can be integrated along a streamline to give

$$\frac{\partial \Phi}{\partial t} + h + \frac{1}{2} \|\mathbf{u}\|^2 = \text{constant} \quad (1.24)$$

which may be recognized as Bernoulli's equation [21].

1.1.3 Speed of Sound

Integrating Eq. (1.13) under the assumption of isentropic flow (i.e. $d\eta = 0$) yields

$$h = \int \frac{dp}{\rho} = \int \left(\frac{dp}{d\rho} \right)_\eta \frac{d\rho}{\rho} \quad (1.25)$$

Pressure, p , has units of force per unit area, and density, ρ , has units of mass per unit volume. From this, $\frac{\partial p}{\partial \rho}$ has units of $\left(\frac{\text{distance}}{\text{time}} \right)^2$ or that of a velocity squared. For convenience, the quantity c is defined as

$$c^2 = \left(\frac{dp}{d\rho} \right)_\eta > 0 \quad (1.26)$$

where the non-negative condition is derived from Eq. (1.8).

From the unit analysis just performed, c is a velocity. To determine the physical meaning of the velocity c , consider Bernoulli's equation expressed in (1.24). Using the expression for h given by Eq. (1.25) in (1.24) gives

$$\frac{\partial \Phi}{\partial t} + \int c^2 \frac{d\rho}{\rho} + \frac{1}{2} (\nabla \Phi \cdot \nabla \Phi) = C \quad (1.27)$$

where C is some constant. Recall that Bernoulli's equation expresses fluid properties along streamlines. Because of this, material coordinates may be used to derive some useful results. From the material expression for continuity [21], it may be written

$$d\rho = -\rho \nabla \cdot \mathbf{u} dt \quad (1.28)$$

Substituting this into the form of Bernoulli's equation given in Eq. (1.27) results in

$$\frac{\partial \Phi}{\partial t} - \int c^2 \nabla^2 \Phi dt + \frac{1}{2} (\nabla \Phi \cdot \nabla \Phi) = C \quad (1.29)$$

where ∇^2 is the Laplace operator. Taking the partial derivative of this with respect to t then gives

$$\frac{\partial^2 \Phi}{\partial t^2} - c^2 \nabla^2 \Phi + \frac{1}{2} \frac{\partial}{\partial t} (\nabla \Phi \cdot \nabla \Phi) = 0 \quad (1.30)$$

which may be rearranged to yield

$$c^2 \nabla^2 \Phi - \frac{\partial^2 \Phi}{\partial t^2} = \frac{1}{2} \frac{\partial}{\partial t} (\nabla \Phi \cdot \nabla \Phi) \quad (1.31)$$

If Eq. (1.31) is expressed in a moving frame attached to a given particle of the flow, then the fluid velocity in the immediate neighborhood is very nearly zero, and the right hand side of this equation may be neglected. Doing so results in the equation

$$c^2 \nabla^2 \Phi - \frac{\partial^2 \Phi}{\partial t^2} = 0 \quad (1.32)$$

which is a version of the three-dimensional wave equation. Thus, small disturbances in Φ propagate through the flow at speed c . Since Eq. (1.32) is linear in Φ , its solutions may be superimposed. Thus, it is satisfied by small disturbances superimposed on the original flow, and c is the speed of propagation of small disturbances through the flow; said differently, c is the local speed of sound [21, 24].

1.1.4 Bernoulli's Equation for Steady Flows

For steady flows, Crocco's theorem (Eq. (1.16)) becomes

$$\left(\frac{1}{2} u_j u_j + h \right)_{,i} - u_j (u_{j,i} - u_{i,j}) = T \eta_{,i} \quad (1.33)$$

Assuming isentropic flow and taking the dot product of this with \mathbf{u} yields

$$u_i u_j u_{j,i} + u_i h_{,i} - u_i u_j (u_{j,i} - u_{i,j}) = 0$$

The third term vanishes since $u_i u_j$ is symmetric and $(u_{j,i} - u_{i,j})$ is skew-symmetric [22]. This then reduces to

$$\mathbf{u} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) + \mathbf{u} \cdot \nabla h = 0$$

along a streamline. Recognizing that $u_i = \frac{dx_i}{dt}$ along a streamline, the above equation may be integrated with respect to t along a streamline, giving

$$\int \frac{1}{2} \nabla (\mathbf{u} \cdot \mathbf{u}) \cdot d\mathbf{x} + \int \nabla h \cdot d\mathbf{x} = \text{constant along a streamline}$$

Upon evaluating the integrals, this yields the steady version of Bernoulli's equation [21]

$$h + \frac{1}{2} \mathbf{u} \cdot \mathbf{u} = h_* = \text{constant on a streamline} \quad (1.34)$$

where h_* is the stagnation or total enthalpy on the streamline. Equation (1.34) gives the total energy per unit mass on a streamline in steady flow and thus gives an expression of the conservation of energy for a fluid particle.

1.1.5 Full (Nonlinear) Potential Equation of Motion

Consider now a steady flow. In this case, the continuity equation and Euler's equations of motion become

$$(\rho u_i)_{,i} = 0 \quad (1.35)$$

and

$$u_i u_{j,i} + \frac{1}{\rho} p_{,j} = 0 \quad (1.36)$$

respectively. Equation (1.25) also reduces to

$$\frac{1}{\rho}p_{,j} = \frac{c^2}{\rho}\rho_{,j} \quad (1.37)$$

Using this in Eq. (1.36) results in

$$u_i u_{j,i} + \frac{c^2}{\rho} \rho_{,j} = 0 \quad (1.38)$$

Taking the dot product of this with \mathbf{u} results in

$$u_j u_i u_{j,i} + \frac{c^2}{\rho} u_j \rho_{,j} = 0 \quad (1.39)$$

Using the product rule, Eq. (1.35) can be rewritten as $u_j \rho_{,j} = -\rho u_{i,i}$. Using this in Eq. (1.39) yields

$$\begin{aligned} c^2 u_{i,i} &= u_j u_i u_{j,i} = u_i \left(\frac{1}{2} u_j u_j \right)_{,i} \\ c^2 \nabla \cdot \mathbf{u} &= \mathbf{u} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) = \mathbf{u} \cdot \nabla \left(\frac{1}{2} \mathbf{u} \cdot \mathbf{u} \right) \end{aligned} \quad (1.40)$$

Note that Eq. (1.40) makes no assumption of irrotationality and so only holds true along a streamline for all inviscid, isentropic flows.

Using Eq. (1.25), the steady version of Bernoulli's equation can be expressed as

$$\int \frac{dp}{\rho} + \frac{1}{2} \mathbf{u} \cdot \mathbf{u} = h_* \quad (1.41)$$

Assuming the flow is isentropic, the differential of Eq. (1.3) is

$$dp = \gamma \rho^{(\gamma-1)} e^{\eta/c_v} d\rho \quad (1.42)$$

Combining this with Eq. (1.25) then yields

$$h = \int \gamma \rho^{(\gamma-2)} e^{\eta/c_v} d\rho = \frac{\gamma}{\gamma-1} \rho^{(\gamma-1)} e^{\eta/c_v} = \frac{\gamma}{\gamma-1} \frac{p}{\rho} \quad (1.43)$$

Equation (1.3) may be manipulated to find

$$\frac{\partial p}{\partial \rho}|_{\eta} = \gamma p / \rho \quad (1.44)$$

Substituting this into the above expression for h gives

$$h = \frac{\gamma}{\gamma - 1} \frac{p}{\rho} = \frac{c^2}{\gamma - 1} \quad (1.45)$$

Using Eq. (1.45) in Eq. (1.34), the latter may be written as

$$c^2 + \frac{\gamma - 1}{2} \mathbf{u} \cdot \mathbf{u} = c_*^2 \quad (1.46)$$

where, again, $*$ denotes the total or stagnation value of a fluid property along a streamline.

Equation (1.46) gives the relationship between the local acoustic speed, c , and the fluid velocity on a streamline in terms of a constant parameter for the streamline, namely, c_* .

Multiplying this relationship by $\nabla \cdot \mathbf{u}$ results in

$$c^2 \nabla \cdot \mathbf{u} + \frac{\gamma - 1}{2} \mathbf{u} \cdot \mathbf{u} \nabla \mathbf{u} = c_*^2 \nabla \cdot \mathbf{u}$$

Using Eq. (1.40), this can be rewritten as

$$\mathbf{u} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) + \frac{\gamma - 1}{2} \mathbf{u} \cdot \mathbf{u} \nabla \mathbf{u} = c_*^2 \nabla \mathbf{u} \quad (1.47)$$

This equation is satisfied by a particle velocity \mathbf{u} in a steady, isentropic flow of a perfect fluid. It is satisfied by the particle velocity rather than velocity field, since the definition of the stagnation speed of sound here and Eq. (1.40) hold only on a streamline.

In the case of irrotational flow, the velocity of the fluid particles \mathbf{u} can be expressed as the gradient of the scalar velocity potential Φ . Applying irrotationality, Eq. (1.47) becomes

$$c_*^2 \nabla^2 \Phi = \nabla \Phi \cdot (\nabla \Phi \cdot \nabla \nabla \Phi) + \frac{\gamma - 1}{2} (\nabla \Phi)^2 \nabla^2 \Phi \quad (1.48)$$

Under the assumption of an irrotational, steady, homentropic, perfect flow, Eq. (1.34) becomes true, not only on a streamline, but across the entire flow field [21]. From (1.45), the stagnation speed of sound, c_* , will also be constant throughout the flow. Equation (1.48) is a form of what is called the full potential equation of motion (c.f. [25–27]). It is the governing equation for the steady motion of an inviscid, isentropic, irrotational, perfect fluid.

1.1.6 Mach Number

If the magnitude of \mathbf{u} is equal to the local acoustic speed (i.e. $|\mathbf{u}| = c$), the flow is said to be sonic. From Eq. (1.46), the sonic speed, c_s , is given by

$$\frac{\gamma + 1}{2} c_s^2 = c_*^2 \quad (1.49)$$

The sonic speed is the speed at which the local fluid speed and the speed of sound are the same. If the flow speed is greater or less than the sonic speed, the flow is called supersonic or subsonic, respectively. From Eqs. (1.46) and (1.49), $|\mathbf{u}|^2 < c_s^2 < c^2$ in the case of subsonic flow and $|\mathbf{u}|^2 > c_s^2 > c^2$ in the case of supersonic flow. Thus, the dimensionless parameter M is defined as

$$M = \frac{|\mathbf{u}|}{c} \quad (1.50)$$

at any point on a streamline. This quantity is called the *Mach number*. Based on the inequalities given above, the Mach number will be greater than, equal to, and less than unity for supersonic, sonic, and subsonic flow, respectively. It should be noted that the value for the local Mach number and, consequently, the designation of the flow, is entirely dependent on the frame of reference chosen.

1.1.7 Linearization to Obtain the Prandtl-Glauert Equation

Equation (1.48) may be solved numerically in a variety of ways. Some of these methods will be discussed subsequently. However, due to the complexity of Eq. (1.48), solving it directly is computationally expensive.

To more easily obtain flow solutions, a linearized version of Eq. (1.48) (the Prandtl-Glauert equation) is more often used. To arrive at the Prandtl-Glauert equation, the full potential equation may be linearized by expressing Φ as a power series in some parameter. The parameter chosen in this case is the ratio of the thickness of the body under consideration to its length. This ratio is denoted by t . If the inclination of the body surface to the freestream is everywhere small, then it is appropriate to assume the perturbations made to the flow by the body are small. This allows for the nonlinear terms in the perturbation velocity components and their derivatives to be neglected in the equations of motion and boundary conditions. In linearizing the equations of motion, it will also be assumed that t is also small and of the same order of magnitude as these surface inclination angles.

Consider now a uniform flow which is continuous, irrotational, and homentropic, with velocity U , constant density ρ_∞ , and constant pressure p_∞ . The speed of sound, c_∞ , of this flow may be obtained from Eq. (1.46), yielding

$$c_\infty^2 + \frac{1}{2}(\gamma - 1)U^2 = c_*^2 \quad (1.51)$$

Combining this with Eq. (1.48) gives

$$c_\infty^2 \nabla^2 \Phi = \nabla \Phi \cdot (\nabla \Phi \cdot \nabla \nabla \Phi) + \frac{1}{2}(\gamma - 1) \left[(\nabla \Phi)^2 - U^2 \right] \nabla^2 \Phi \quad (1.52)$$

which describes the motion of an adiabatic, inviscid, steady flow originating in a uniform flow with velocity U .

To fully define this flow, a right-handed orthogonal Cartesian system is defined that is fixed relative to the body. The coordinate system is oriented so that the x -axis is parallel

to the direction of the freestream. The velocity potential, Φ , may now be expanded in the series

$$\Phi = U(x + \phi_1 t + \phi_2 t^2 + \phi_3 t^3 + \dots) \quad (1.53)$$

where Ux represents the freestream contribution to the velocity potential, and all other terms are perturbations due to the presence of the body. It is assumed that this series is both convergent and represents a solution to Eq. (1.52). Substituting this series into Eq. (1.52) and equating the coefficient of successive powers of t yields the series of equations [21]

$$\nabla^2 \phi_1 - M_\infty^2 \frac{\partial^2 \phi_1}{\partial x^2} = 0 \quad (1.54)$$

$$\nabla^2 \phi_2 - M_\infty^2 \frac{\partial^2 \phi_2}{\partial x^2} = M_\infty^2 \frac{\partial}{\partial x} (\nabla \phi_1)^2 + (\gamma - 1) M_\infty^2 \frac{\partial \phi_1}{\partial x} \nabla^2 \phi_1 \quad (1.55)$$

$$\begin{aligned} \nabla^2 \phi_3 - M_\infty^2 \frac{\partial^2 \phi_3}{\partial x^2} = M_\infty^2 \left[4 \frac{\partial}{\partial x} (\nabla \phi_1 \cdot \nabla \phi_2) + \nabla \phi_1 \cdot (\nabla \phi_1 \cdot \nabla \nabla \phi_1) \right] + \\ (\gamma - 1) M_\infty^2 \left[\frac{\partial \phi_1}{\partial x} \nabla^2 \phi_2 + \frac{\partial \phi_2}{\partial x} \nabla^2 \phi_1 + \frac{1}{2} (\nabla \phi_1)^2 \nabla^2 \phi_1 \right] \end{aligned} \quad (1.56)$$

and so forth, where $M_\infty = U/c_\infty$ is the Mach number of the uniform onset flow. The equating of coefficients done to arrive at Eqs. (1.54) - (1.56) is performed to relate orders of magnitude of the parameter t to establish their influence on the solution. Note that Eq. (1.54) is the Prandtl-Glauert equation

$$\nabla^2 \phi_1 - M_\infty^2 \frac{\partial^2 \phi_1}{\partial x^2} = 0 \quad (1.57)$$

Assuming that the perturbation potentials beyond ϕ_1 (i.e. ϕ_2, ϕ_3 , etc.) do not contribute significantly to the flow (valid for $t \ll 1$), the governing equation for the perturbation flow is simply

$$(1 - M_\infty^2) \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0 \quad (1.58)$$

Equation (1.58) is the form of the Prandtl-Glauert equation most commonly seen.

It is critical to remember that the Prandtl-Glauert equation emerges only as the first term in a series [28]. If the series does not converge, then the solution of its first term is meaningless. Even if the series converges slowly, the solution arising from the first term may not provide a good estimate of the exact solution. Clearly, if t is not small, then the series will likely diverge. How convergence depends on the Mach number is examined in the following section.

Alternate Linearization

There is another method for deriving the Prandtl-Glauert equation that will be considered in this section. Consider a uniform freestream with constant velocity, \mathbf{U} , density, ρ_∞ , and pressure, p_∞ . Letting \mathbf{u}' be the velocity perturbation from \mathbf{U} , the fluid velocity may be written as

$$\mathbf{u} = \mathbf{U} + \mathbf{u}' \quad (1.59)$$

Assume that squares, products, and higher powers of \mathbf{u}' , $\rho - \rho_\infty$, $p - p_\infty$, and $\eta - \eta_\infty$ are small enough to be neglected (i.e. small perturbations). The continuity equation can then be written as

$$\frac{\partial(\rho - \rho_\infty)}{\partial t} + \nabla \cdot (\rho \mathbf{u} - \rho_\infty \mathbf{u}) = 0$$

$$\frac{\partial(\rho - \rho_\infty)}{\partial t} + \nabla \cdot (\rho(\mathbf{U} + \mathbf{u}') - \rho_\infty(\mathbf{U} + \mathbf{u}')) = 0$$

$$\frac{D}{Dt} \left(\frac{\rho - \rho_\infty}{\rho_\infty} \right) + \nabla \cdot \mathbf{u}' = 0 \quad (1.60)$$

where the material derivative is expressed (assuming small perturbations) as

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{U} \cdot \nabla \quad (1.61)$$

Under the small perturbation assumptions, Euler's equation of motion (Eq. (1.2)) becomes

$$\frac{\partial(\mathbf{U} + \mathbf{u}')}{\partial t} + (\mathbf{U} + \mathbf{u}') \cdot \nabla (\mathbf{U} + \mathbf{u}') + \frac{1}{\rho_\infty} \nabla(p - p_\infty) = 0$$

$$\frac{\partial \mathbf{u}'}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{u}' + \frac{1}{\rho_\infty} \nabla(p - p_\infty) = 0$$

$$\frac{D\mathbf{u}'}{Dt} + \nabla \left(\frac{p - p_\infty}{\rho_\infty} \right) = 0 \quad (1.62)$$

and the expression of isentropic flow becomes

$$\frac{D}{Dt} (\eta - \eta_\infty) = 0 \quad (1.63)$$

The equation of state is assumed to be of the form $p = f(\rho, \eta)$. By expanding this in a Taylor series and retaining only linear terms, the linearized equation of state is found to be

$$p - p_\infty = \left(\frac{\partial p}{\partial \rho} \right)_{\eta_\infty} (\rho - \rho_\infty) + \left(\frac{\partial p}{\partial \eta} \right)_{\rho_\infty} (\eta - \eta_\infty) = c_\infty^2 (\rho - \rho_\infty) + \left(\frac{\partial p}{\partial \eta} \right)_{\rho_\infty} (\eta - \eta_\infty) \quad (1.64)$$

where c_∞ is the speed of sound in the freestream.

Assuming steady flows, the linearized continuity equation becomes

$$\mathbf{U} \cdot \nabla (\rho - \rho_\infty) + \rho_\infty \nabla \cdot \mathbf{u}' = 0 \quad (1.65)$$

and the linearized equations of motion and entropy equations are, respectively,

$$\rho_\infty \mathbf{U} \cdot \nabla \mathbf{u}' + \nabla (p - p_\infty) = 0 \quad (1.66)$$

$$\mathbf{U} \cdot \nabla (\eta - \eta_\infty) = 0 \quad (1.67)$$

The quantities ρ , p , and η may be eliminated from these equations first by rewriting Eq. (1.66) using Eq. (1.64)

$$\rho_{\infty} U_i u'_{j,i} + \left[c_{\infty}^2 (\rho - \rho_{\infty}) + \left(\frac{dp}{d\eta} \right)_{\rho_{\infty}} (\eta - \eta_{\infty}) \right]_{,j} = 0$$

Taking the inner product of this with \mathbf{U} results in

$$\rho_{\infty} U_j U_i u'_{j,i} + U_j \left[c_{\infty}^2 (\rho - \rho_{\infty}) + \left(\frac{dp}{d\eta} \right)_{\rho_{\infty}} (\eta - \eta_{\infty}) \right]_{,j} = 0$$

Because of Eq. (1.67), this reduces to

$$\rho_{\infty} U_j U_i u'_{j,i} + c_{\infty}^2 U_j \rho_{,j} = 0$$

Then using Eq. (1.65), this may be written as

$$\rho_{\infty} U_j U_i u'_{j,i} - c_{\infty}^2 \rho_{\infty} u'_{i,i} = 0$$

which becomes

$$u'_{i,i} - U_j U_i u'_{j,i} / c_{\infty}^2 = 0 \quad (1.68)$$

Since \mathbf{U} is constant in space, this becomes

$$\nabla \cdot ([I] - \mathbf{U}\mathbf{U}^T / c_{\infty}^2) \cdot \mathbf{u}' = 0 \quad (1.69)$$

or

$$\nabla \cdot \mathbf{w} = 0 \quad (1.70)$$

where $\mathbf{w} = ([I] - \mathbf{U}\mathbf{U}^T / c_{\infty}^2) \cdot \mathbf{u}'$. Let

$$[B] = [I] - \mathbf{U}\mathbf{U}^T / c_{\infty}^2 \quad (1.71)$$

in which case

$$\mathbf{w} = [B]\mathbf{u}' \quad (1.72)$$

Equation (1.70) reveals that \mathbf{w} is the flux of a conserved quantity in linearized theory [21]. Denoting $\mathbf{u}' = [u, v, w]$ and recalling the freestream is aligned with the x -axis, \mathbf{w} may be written as

$$\mathbf{w} = [(1 - M_\infty^2)u, v, w] \quad (1.73)$$

The vector \mathbf{w} is called the *reduced current velocity* [21, 29, 30]. The physical meaning of \mathbf{w} will be examined in the following section.

If the flow is assumed to be irrotational, then \mathbf{u}' can be expressed as the gradient of a potential function $\mathbf{u}' = \nabla\phi$, where ϕ is the *perturbation potential*. From Eq. (1.70), the equation governing ϕ is then simply

$$(1 - M_\infty^2) \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0 \quad (1.74)$$

This is the same as Eq. (1.58) obtained previously by performing an order of magnitude comparison in t . Note the previous derivation using a series expansion for Φ is more restrictive and more rigorous than the one just performed. Eq. (1.74) is ultimately justified by the more rigorous derivation.

In arriving at this equation, it was assumed that products and powers of the perturbation quantities may be neglected. However, when $M_\infty \approx 1$, certain neglected terms are of the same order of magnitude as the retained terms and so may not be neglected [31]. Because of this, the Prandtl-Glauert equation is not valid near $M_\infty = 1$. Retaining these extra terms leads to what is known as the transonic small disturbance equation [31, 32]. A discussion of this equation is beyond the scope of the current work. However, it must be recognized that the assumptions used to arrive at the Prandtl-Glauert equation cease to be valid near $M_\infty = 1$.

1.1.8 Linearized Flow Properties

Once ϕ is determined and assuming the flow is homentropic (i.e. $\eta = \eta_\infty$), the pressure at any point in the flow can be calculated using Bernoulli's equation, restated here as

$$\int_{p_\infty}^p \frac{dp}{\rho} + \frac{1}{2} \mathbf{u} \cdot \mathbf{u} = \frac{1}{2} U^2 \quad (1.75)$$

Assuming small perturbations in pressure and density from the freestream, the integrand in this equation can be written as a Taylor series in $(p - p_\infty)$

$$\int_{p_\infty}^p \left[\frac{1}{\rho_\infty} - \frac{1}{\rho_\infty^2} \left(\frac{d\rho}{dp} \right)_{\eta_\infty} (p - p_\infty) + \dots \right] dp \quad (1.76)$$

Upon evaluating the integral, Bernoulli's equation becomes

$$\frac{p - p_\infty}{\rho_\infty} - \frac{1}{2c_\infty^2} \left(\frac{p - p_\infty}{\rho_\infty} \right)^2 + \dots = \frac{1}{2} \mathbf{U} \cdot \mathbf{U} - \frac{1}{2} \mathbf{u} \cdot \mathbf{u} = -\mathbf{U} \cdot \mathbf{u}' - \frac{1}{2} \mathbf{u}' \cdot \mathbf{u}' \quad (1.77)$$

which may be rearranged to produce

$$\frac{p - p_\infty}{\rho_\infty} = -\mathbf{U} \cdot \mathbf{u}' - \frac{1}{2} \mathbf{u}' \cdot \mathbf{u}' + \frac{1}{2c_\infty^2} \left(\frac{p - p_\infty}{\rho_\infty} \right)^2 + \dots \quad (1.78)$$

Since the left-hand term of this equation appears on the right hand, this equation may be applied recursively to find

$$\frac{p - p_\infty}{\rho_\infty} = -\mathbf{U} \cdot \mathbf{u}' - \frac{1}{2} \mathbf{u}' \cdot \mathbf{u}' + \frac{1}{2c_\infty^2} \left(-\mathbf{U} \cdot \mathbf{u}' - \frac{1}{2} \mathbf{u}' \cdot \mathbf{u}' + \frac{1}{2c_\infty^2} (\dots)^2 + \dots \right)^2 + \dots \quad (1.79)$$

Upon neglecting higher-order terms, this becomes

$$\frac{p - p_\infty}{\rho_\infty} \approx -\mathbf{U} \cdot \mathbf{u}' - \frac{1}{2} \mathbf{u}' \cdot \mathbf{u}' + \frac{1}{2c_\infty^2} \left(\mathbf{U} \cdot \mathbf{u}' + \frac{1}{2} \mathbf{u}' \cdot \mathbf{u}' \right)^2 \quad (1.80)$$

$$\frac{p - p_\infty}{\rho_\infty} \approx -U_i v_i - \frac{1}{2} v_i v_i + \frac{1}{2c_\infty^2} \left(U_i U_j v_i v_j + U_i v_j v_i v_j + \frac{1}{4} v_i v_i v_j v_j \right)$$

Neglecting cubic powers of the components \mathbf{u}' and higher then yields

$$\frac{p - p_\infty}{\rho_\infty} \approx -U_i v_i - \frac{1}{2} \left(v_i v_i - \frac{1}{2c_\infty^2} U_i U_j v_i v_j \right)$$

$$\frac{p - p_\infty}{\rho_\infty} \approx -U_i v_i - \frac{1}{2} \left(v_i - \frac{1}{2c_\infty^2} U_i U_j v_j \right) v_i = -U_i v_i - \frac{1}{2} w_i v_i$$

$$\frac{p - p_\infty}{\rho_\infty} \approx -\mathbf{U} \cdot \mathbf{u}' - \frac{1}{2} \mathbf{u}' \cdot \mathbf{w} \quad (1.81)$$

This is called the quadratic approximation to Bernoulli's equation. Dropping all nonlinear terms yields the linear approximation

$$\frac{p - p_\infty}{\rho_\infty} \approx -\mathbf{U} \cdot \mathbf{u}' \quad (1.82)$$

By means of either of these equations, the pressure distribution in the flow can be determined from the linear perturbation velocities. The isentropic pressure relation is also often used to calculate pressures [33, 34].

Similarly, the changes in density can be approximated by writing ρ as a Taylor series in $p - p_\infty$

$$\rho = \rho_\infty + \left(\frac{\partial \rho}{\partial p} \right)_{\eta_\infty} (p - p_\infty) + \dots = \rho_\infty + (p - p_\infty) / c_\infty^2 + \dots \quad (1.83)$$

Applying Eq. (1.82) to this and neglecting higher-order terms results in

$$\rho - \rho_\infty \approx -\rho_\infty \mathbf{U} \cdot \mathbf{u}' / c_\infty^2 \quad (1.84)$$

The perturbation mass flux may be determined by multiplying Eq. (1.84) by \mathbf{u} and rearranging

$$\rho (\mathbf{U} + \mathbf{u}') = \rho_\infty (1 - \mathbf{U} \cdot \mathbf{u}' / c_\infty^2) (\mathbf{U} + \mathbf{u}')$$

Subtracting $\rho_\infty \mathbf{U}$ from both sides gives

$$\rho \mathbf{u} - \rho_\infty \mathbf{U} = \rho_\infty (\mathbf{U} - \mathbf{U} + \mathbf{u}' - \mathbf{U} \mathbf{U} \cdot \mathbf{u}' / c_\infty^2 + \dots) = \rho_\infty \mathbf{w} + \dots$$

Thus, to a linear approximation

$$\mathbf{w} = (\rho \mathbf{u} - \rho_\infty \mathbf{U}) / \rho_\infty \quad (1.85)$$

Thus, \mathbf{w} is proportional to the linearized perturbation mass flux. This is a useful result for applying boundary conditions in panel methods.

1.2 Characteristic Surfaces

Before discussing how the Prandtl-Glauert equation may be solved for realistic configurations, it will be useful to examine some of its qualitative aspects. In particular, the characteristic surfaces of both the full potential equation (Eq. (1.52)) and the Prandtl-Glauert equation will be examined. Understanding the characteristic surfaces of the full potential equation gives insight into the behavior of the Prandtl-Glauert equation.

1.2.1 Characteristic Surfaces of the Full Potential Equation

Consider Eq. (1.40) subject to the constraint $\nabla \times \mathbf{u} = 0$ (i.e. irrotationality). This constraint may be combined with Eq. (1.40) as in

$$c^2 \nabla \cdot \mathbf{u} - \mathbf{u} \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) + \boldsymbol{\lambda} \cdot \nabla \times \mathbf{u} = 0 \quad (1.86)$$

where $\boldsymbol{\lambda}$ is a constraint vector to be determined. Note the dot product with λ enforces the irrotationality constraint in a weaker form

It will be shown here that, in a continuous flow, the normal derivative of the velocity may be discontinuous at certain surfaces while the tangential derivatives are still continuous.

In such cases, the equations of motion must still be satisfied, and so this is only possible if the equations of motion can be expressed only in terms of the tangential derivatives, the normal derivatives being absent. Let S denote such a surface having normal vector \mathbf{n} . The vector $\boldsymbol{\lambda}$ given previously is determined by requiring that Eq. (1.86) contains only tangential derivatives of \mathbf{u} on S .

To determine what S is, Eq. (1.86) can be written as

$$(c^2\delta_{ij} - u_j u_i + \varepsilon_{ijk}\lambda_k) u_{j,i} = 0 \quad (1.87)$$

Recall that the derivative of a function f in the direction of \mathbf{n} is given by $n_i f_{,i}$. For clarity of interpretation, Eq. (1.87) may be rewritten as

$$\delta_{jm} [(c^2\delta_{ij} - u_j u_i + \varepsilon_{ijk}\lambda_k) u_{m,i}] = 0 \quad (1.88)$$

which is the trace of a 2-tensor composed of directional derivatives of the components of \mathbf{u} , the term within the parentheses representing a directional specification for any fixed j . Because the term within the parentheses represents a directional specification, the governing equations will contain only tangential derivatives on S if the term within the parentheses is perpendicular to \mathbf{n} . This may be written as

$$(c^2\delta_{ij} - u_j u_i + \varepsilon_{ijk}\lambda_k) n_i = 0 \quad (1.89)$$

for any given j . Distributing the dot product, this yields

$$\begin{aligned} c^2 n_j - u_j u_i n_i + \varepsilon_{ijk} n_i \lambda_k &= 0 \\ c^2 \mathbf{n} - \mathbf{u}(\mathbf{u} \cdot \mathbf{n}) + \boldsymbol{\lambda} \times \mathbf{n} &= 0 \end{aligned} \quad (1.90)$$

Recognizing that $\boldsymbol{\lambda} \times \mathbf{n}$ will always be perpendicular to \mathbf{n} , this equation will only have a solution if $c^2 \mathbf{n} - \mathbf{u}(\mathbf{u} \cdot \mathbf{n})$ is also perpendicular to \mathbf{n} . This may be written as

$$c^2 \mathbf{n} \cdot \mathbf{n} - \mathbf{u} (\mathbf{u} \cdot \mathbf{n}) \cdot \mathbf{n} = 0$$

$$c^2 = (\mathbf{u} \cdot \mathbf{n})^2 \tag{1.91}$$

This is a condition for the existence of S . Since \mathbf{n} is a unit vector, it must be that

$$(\mathbf{u} \cdot \mathbf{n})^2 \leq |\mathbf{u}|^2 \tag{1.92}$$

Thus

$$c^2 \leq |\mathbf{u}|^2 \tag{1.93}$$

This condition states that the surface S only exists for flow situations in which $M^2 \geq 1$, that is, for locally supersonic flows. The surfaces where Eq. (1.91) is satisfied are referred to as *characteristic surfaces* [21]. From Eq. (1.91), on a characteristic surface, the normal component of velocity is equal to the local speed of sound.

These characteristic surfaces correspond to shock waves in supersonic flow [21]. At these characteristic surfaces, the differential equations of motion are no longer valid. However, conservation of mass, momentum, and energy may be used to analyze the changes in the flow as it passes through such surfaces [21, 24, 34]. Performing such an analysis reveals that the flow may be compressed or expanded as it passes through a shock. However, an expansive shock yields a decrease in entropy, which is not physically possible [21, 34, 35]; hence the appearance of expansion fans. Unfortunately, such expansion fans do not appear in solutions to Eqs. (1.48) or (1.58) without the addition of artificial viscosity. This is one weakness to analyzing supersonic flows using purely inviscid methods. It is possible to use numerical viscosity or other means to suppress expansive shocks in certain applications of inviscid flow theory (for example, see [25, 32, 36–40]); however, no such options exist for linear panel methods.

Because of the jump in entropy across a shock, it is not generally true that a homentropic flow upstream of a shock results in a homentropic flow downstream of the shock. Similarly, irrotationality of a flow is not generally preserved after passing through a shock [21]. This is another weakness of panel methods, resulting in their accuracy being limited to flows without strong shocks (i.e. slender/thin bodies moving at low supersonic Mach numbers). This is typically justified, especially for preliminary design, as it is not desirable to have strong shocks appear on aircraft at cruise [39].

1.2.2 Characteristic Surfaces of the Prandtl-Glauert Equation

The condition for the existence of characteristic surfaces of the Prandtl-Glauert equation is obtained by linearizing Eq. (1.91), yielding

$$c_\infty^2 = (\mathbf{U} \cdot \mathbf{n})^2 \quad (1.94)$$

Once again, this shows that real characteristic surfaces will only exist in supersonic flow. This fits with the hyperbolic nature of the supersonic Prandtl-Glauert equation, which may be written as

$$-B^2 \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0 \quad (1.95)$$

where $B = \sqrt{M_\infty^2 - 1}$. In subsonic flow, the Prandtl-Glauert equation is elliptic, and characteristic surfaces do not exist. For a discussion of the differences between hyperbolic and elliptic PDEs, see [41].

In linearized flow, c_∞ is constant for the whole fluid. Because of this, any disturbance introduced at a point (x_d, y_d, z_d) in the flow will propagate outward at a speed of c_∞ . This can be viewed as a spherical wavefront whose radius is growing at a rate of c_∞ . However, these disturbances are also being swept downstream at a rate of U . If $U > c_\infty$ (i.e. $M_\infty > 1$), then the disturbances are being swept downstream faster than they are propagated outward. This means the spherical wavefronts described are enveloped by a cone whose equation is

$$\{(x, y, z) \mid (x - x_d)^2 = B^2 [(y - y_d)^2 + (z - z_d)^2]\} \quad (1.96)$$

This cone is called the *Mach cone*. The half angle of the Mach cone is equal to the Mach angle, given by [21]

$$\mu = \sin^{-1} \frac{1}{M_\infty} \quad (1.97)$$

Note that two cones, one extending upstream from (x_d, y_d, z_d) and one extending downstream, satisfy Eq. (1.96). The upstream Mach cone encompasses the region of all points which may influence the point (x_d, y_d, z_d) and the downstream Mach cone encompasses the region of all points which are influenced by the point. These regions are called the *domain of dependence* and *domain of influence*, respectively. These domains play a significant role in the solution of the supersonic Prandtl-Glauert equation using panel methods.

The downstream Mach cone, being a characteristic surface, represents an infinitely weak shock wave in linearized theory [21]. On these surfaces, discontinuities in velocity and pressure may appear. This is to be expected, as hyperbolic PDEs admit piecewise-analytic solutions [41].

1.2.3 Sub- and Superinclined Surfaces

Having discussed the characteristic surfaces of the Prandtl-Glauert equation, it is appropriate to introduce here the concept of subinclined and superinclined surfaces in supersonic flow. Recall that the effect of any disturbance introduced at a given point is limited to the region surrounded by the Mach cone downstream of that point. Consider now that this point lies on a surface S with normal vector \mathbf{n} . If at a point P , S is inclined to the freestream vector at an angle greater than the Mach angle, μ , then the Mach cone originating at point P will not intersect S . In this case, S is called *superinclined*. However, if S is inclined to the freestream vector at an angle less than the Mach angle, the the Mach cone originating at point P will intersect S , and S is called *subinclined* at P . Examples of these two types of surfaces are shown in Fig. 1.1.

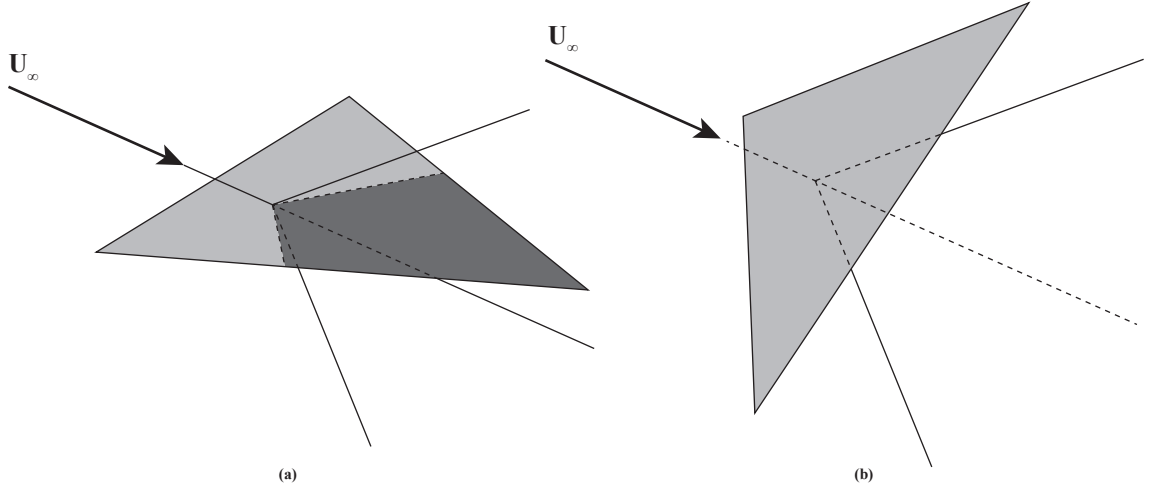


Fig. 1.1: Examples of (a) sub- and (b) superinclined surfaces showing the intersection of the domain of dependence with the surface.

If \mathbf{n} is the normal vector to S , then S is superinclined if

$$|n_x| > \sin \mu \quad (1.98)$$

and subinclined if

$$|n_x| < \sin \mu \quad (1.99)$$

assuming the freestream velocity is aligned with the x -axis. The case where $|n_x| = \sin \mu$ is called *Mach-inclined*.

Subinclined and superinclined surfaces are handled differently within a panel method. This work focuses primarily on flow solutions over subinclined surfaces, though information on superinclined surfaces is given throughout. Superinclined surfaces, along with the difficulties in their implementation, are discussed primarily in Section 2.2.

1.3 Method of Green's Functions

While the Prandtl-Glauert equation is a remarkably simple equation, especially compared to the full potential equation, obtaining solutions to the Prandtl-Glauert equation is still nontrivial. While there are many ways to do this, this work focuses on panel methods, which arise quite naturally from the method of Green's functions [41, 42].

Most technical readers will be familiar with the concept of finite volume methods so often used in computational fluid dynamics. Such methods involve discretizing the domain of interest and applying a finite difference approximation of the governing equations to each discrete domain element. At the boundary of the domain, boundary conditions are enforced again using finite-difference approximations. This results in a large, sparse system of equations which may be used to determine the properties of interest at each point in the domain. It may be said that such methods solve a partial differential equation approximately using exact (in a finite-difference sense) boundary conditions.

On the other hand, the method of Green's functions solves the partial differential equation exactly while only enforcing boundary conditions approximately, similar to the method of eigenfunctions [41, 42]. However, unlike the method of eigenfunctions, the boundary conditions are approximated by assuming certain distributions (polynomial, etc.) for the boundary conditions, rather than using infinite series.

The method of Green's functions is most easily illustrated by an example from another field of physics: electrostatics. Recall that electrostatic potentials also satisfy Laplace's equation [43]. Consider a single point charge isolated in space. From electrostatics, the electric field induced at any point in space by this point charge may be calculated as a function of the relative position of the point of interest and the magnitude of the charge. Now consider two point charges isolated from one another by some finite distance. Since Laplace's equation is linear, the induced field at any point may be found by superimposing the fields induced by each of the two charges.

Consider now the electric field inside of a charged box of zero thickness, where the electric charge on each point of the surface of the box is known. Since the box may be

considered an infinite collection of point charges, the same principle applies as before. The induced potential inside the box may be found by integrating the induced potentials from each point on the box. Since the field induced by a single point charge satisfies Laplace's equation and Laplace's equation is linear, the potential inside the box will exactly satisfy the governing equation at all points. Having started with a known distribution of charges, the boundary conditions are also exactly satisfied.

Consider now how the above method may be generalized. The starting point is a linear partial differential equation. A solution to that differential equation representing the influence of a point disturbance is then determined. This fundamental solution is called the Green's function for that PDE [42]. The fundamental solution is then integrated over the boundary of the region of interest such that the resulting distribution satisfies the desired boundary conditions, either exactly or approximately.

Inversely, the influence of the sources distributed on the boundary at a given set of points may be calculated as a function of the unknown boundary values. Specifying some condition for the induced potentials at each of these points will produce a system of equations which may be used to solve for the boundary values. This is the approach taken within panel methods. Based on a knowledge of the potential outside (or inside) the configuration of interest, the distribution of boundary values on the configuration is determined to satisfy those given conditions.

The application of the method of Green's functions differs somewhat between subsonic and supersonic flows. The resulting integral equation is very similar for both. However, the differences in derivation are significant enough to warrant examining them separately. In both cases, the divergence theorem is used to obtain a modified version of Green's theorem. The derivations as presented here are taken primarily from Ward [21].

1.3.1 General Solution for Subsonic Flows

For an irrotational, source-free, subsonic flow, the governing equations are

$$\nabla \cdot \mathbf{w} = 0$$

$$\mathbf{w} = [B]\mathbf{u}'$$

$$\mathbf{u}' = \nabla \phi$$

repeated from previously. Consider two subsonic flows, defined by ϕ_1 , \mathbf{w}_1 , and \mathbf{u}'_1 , and ϕ_2 , \mathbf{w}_2 , and \mathbf{u}'_2 , respectively, having the same Mach number. Since the Mach numbers are the same, it must be that

$$\mathbf{u}'_1 \cdot \mathbf{w}_2 = \mathbf{u}'_1 [B] \mathbf{u}'_2 = \mathbf{w}_1 \cdot \mathbf{u}'_2 \quad (1.100)$$

Consider now a closed surface S with outward normal vector \mathbf{n} (i.e. pointing outward from the flow region). Applying the divergence theorem to the vector $\phi_1 \mathbf{w}_2$ over S yields

$$\iint_S \phi_1 \mathbf{w}_2 \cdot \mathbf{n} dS = \iiint_V \nabla \cdot (\phi_1 \mathbf{w}_2) dV = \iiint_V \nabla \phi_1 \cdot \mathbf{w}_2 + \phi_1 \nabla \cdot \mathbf{w}_2 dV \quad (1.101)$$

This may also be done for the vector $\phi_2 \mathbf{w}_1$, resulting in

$$\iint_S \phi_2 \mathbf{w}_1 \cdot \mathbf{n} dS = \iiint_V \nabla \phi_2 \cdot \mathbf{w}_1 + \phi_2 \nabla \cdot \mathbf{w}_1 dV \quad (1.102)$$

Subtracting these two and applying Eq. (1.100) gives

$$\iint_S (\phi_1 \mathbf{w}_2 - \phi_2 \mathbf{w}_1) \cdot \mathbf{n} dS = \iiint_V \phi_1 \nabla \cdot \mathbf{w}_2 - \phi_2 \nabla \cdot \mathbf{w}_1 dV \quad (1.103)$$

which is analogous to Green's second identity (c.f. Eq. (II) on page 215 of [43]).

Now assume that ϕ_1 and the corresponding velocity and reduced current velocity define the subsonic flow of interest and satisfy the governing equations for an irrotational, source-free, subsonic flow. In addition, let

$$\phi_2 = \frac{1}{R_\beta} \quad (1.104)$$

where

$$R_\beta = \sqrt{(x - \xi)^2 + (1 - M_\infty^2) [(y - \eta)^2 + (z - \zeta)^2]} \quad (1.105)$$

where (x, y, z) are the coordinates of a point \mathbf{P} in the flowfield and (ξ, η, ζ) are the coordinates of the integration point \mathbf{Q} .

Dropping the subscripts, Eq. (1.103) then becomes

$$\iint_S (\phi[B] \nabla \frac{1}{R_\beta} - \frac{1}{R_\beta} [B] \nabla \phi) \cdot \mathbf{n} dS = \iiint_V \phi \nabla [B] \nabla \frac{1}{R_\beta} dV \quad (1.106)$$

Letting

$$\tilde{\mathbf{n}} = [B] \mathbf{n} \quad (1.107)$$

where $\tilde{\mathbf{n}}$ is called the *conormal vector*, this may be written in the more condensed form

$$\iint_S \phi \frac{\partial}{\partial \tilde{n}} \left(\frac{1}{R_\beta} \right) - \frac{1}{R_\beta} \frac{\partial \phi}{\partial \tilde{n}} dS = \iiint_V \phi \nabla [B] \nabla \frac{1}{R_\beta} dV \quad (1.108)$$

where

$$\frac{\partial}{\partial \tilde{n}} = \mathbf{n}^T [B] \nabla = \tilde{\mathbf{n}}^T \cdot \nabla \quad (1.109)$$

Here, $\frac{\partial}{\partial \tilde{n}}$ is called the *conormal derivative*.

As defined, the integrands in Eq. (1.108) are singular at \mathbf{P} . To overcome this, \mathbf{P} may be excluded from V by adding to its boundary a small sphere S_1 of radius ϵ centered at \mathbf{P} [21]. The interior of this sphere is excluded from V , and so the volume integral in Eq.

(1.108) may be evaluated if the corresponding surface integral is taken over S and S_1 . The surface integral over S_1 is given by

$$\iint_{S_1} \phi \frac{\partial}{\partial \tilde{n}} \left(\frac{1}{R_\beta} \right) - \frac{1}{R_\beta} \frac{\partial \phi}{\partial \tilde{n}} dS_1$$

$$\iint_{S_1} -\phi \frac{\beta^2 (\mathbf{Q} - \mathbf{P}) \cdot \mathbf{n}}{R_\beta^3} - \frac{1}{R_\beta} \frac{\partial \phi}{\partial \tilde{n}} dS_1$$

Note that \mathbf{n} points into the region bounded by S_1 . These integrals evaluate to [21]

$$4\pi \bar{\phi} - \frac{4\pi\epsilon}{M_\infty} \sinh^{-1} \left(\frac{M_\infty}{\beta} \right) \frac{\partial \bar{\phi}}{\partial \tilde{n}} \quad (1.110)$$

where the bars denote average values on S_1 . Taking the limit as $\epsilon \rightarrow 0$ (i.e. making the sphere S_1 vanishingly small), these simply become

$$4\pi \phi(\mathbf{P}) \quad (1.111)$$

It may also be shown that (see Appendix A), for $\mathbf{P} \neq \mathbf{Q}$ (which is true outside of S_1)

$$\nabla[B] \nabla \frac{1}{R_\beta} = 0 \quad (1.112)$$

Combining these results with Eq. (1.108) yields

$$\iint_S \phi \frac{\partial}{\partial \tilde{n}} \left(\frac{1}{R_\beta} \right) - \frac{1}{R_\beta} \frac{\partial \phi}{\partial \tilde{n}} dS + 4\pi \phi(\mathbf{P}) = 0 \quad (1.113)$$

or

$$\phi(\mathbf{P}) = \frac{1}{4\pi} \iint_S \frac{1}{R_\beta} \frac{\partial \phi}{\partial \tilde{n}} - \phi \frac{\partial}{\partial \tilde{n}} \left(\frac{1}{R_\beta} \right) dS \quad (1.114)$$

Equation (1.114) reveals the remarkable result that for an irrotational, source-free, subsonic flow, the velocity potential, ϕ , at any point in the flow may be found as a function of its value and conormal derivative on the surface bounding the flowfield. That this is uniquely

specified may be shown (see Section 2.4 in [21]). This surface-only dependence is what lends panel methods their speed.

1.3.2 General Solution for Supersonic Flows

Deriving the general solution for supersonic flows is very similar to the process for subsonic flows, except that special care must be taken to deal with the singularities appearing on the Mach cones.

As with the subsonic derivation, consider two irrotational, source-free flows for which Eq. (1.103) holds. Again, ϕ_1 , \mathbf{w}_1 , and \mathbf{u}'_1 describe the flow of interest and satisfy the governing equations. Additionally, $\mathbf{P} = (x, y, z)$ is the point of interest in the flow and $\mathbf{Q} = (\xi, \eta, \zeta)$ is the integration point. Let ϕ_2 be given by

$$\phi_2 = \begin{cases} \frac{1}{R_B}, & R_B^2 > 0, x > \xi \\ 0, & \text{otherwise} \end{cases} \quad (1.115)$$

where

$$R_B^2 = (x - \xi)^2 - (M_\infty^2 - 1) [(y - \eta)^2 + (z - \zeta)^2] \quad (1.116)$$

Since ϕ_2 is zero for a large portion of the fluid region (recall the domains of dependence and influence described previously), the surface and volume integrals need only be taken over a limited region. This region will be bounded by the part of the upstream Mach cone ($R_B = 0$ and $x > \xi$) downstream of S (this portion of the Mach cone is denoted by Γ) and the part of S inside of Γ as shown in Fig. 1.2. In addition, the point \mathbf{P} is removed from V by introducing the surface S_1 encompassed by Γ and offset from \mathbf{P} by δ .

Applying these definitions, Eq. (1.103) becomes, for supersonic flows

$$\iint_S (\phi[B] \nabla \frac{1}{R_B} - \frac{1}{R_B} [B] \nabla \phi) \cdot \mathbf{n} dS = \iiint_V \phi \nabla [B] \nabla \frac{1}{R_B} dV \quad (1.117)$$

With Eq. (1.107), this may be simplified to

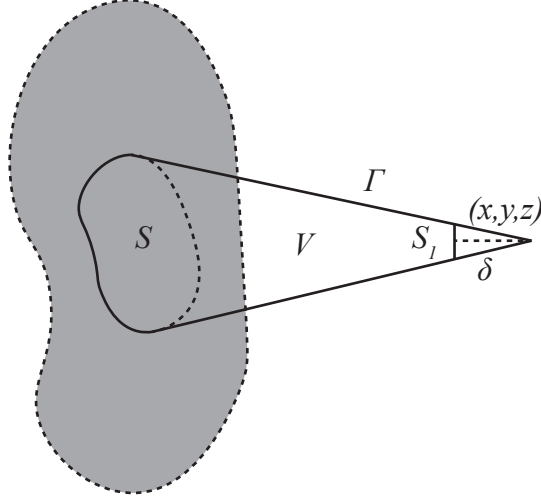


Fig. 1.2: Region of integration for the supersonic Green's theorem.

$$\iint_S \phi \frac{\partial}{\partial \tilde{n}} \frac{1}{R_B} - \frac{1}{R_B} \frac{\partial \phi}{\partial \tilde{n}} dS = \iiint_V \phi \nabla[B] \nabla \frac{1}{R_B} dV \quad (1.118)$$

Since $R_B = 0$ on Γ , the surface integrals diverge. To avoid this, finite-part integrals are used (for a description of finite-part integrals in this context, see Section 3.4 in [21]). Doing so, both integrals over Γ vanish. The source integral over S converges, while the finite part of the doublet integral over S may be retained. As $\delta \rightarrow 0$, the source integral over S_1 vanishes, while the finite part of the doublet integral is found to be [21]

$$\iint_{S_1}^* \phi \frac{\partial}{\partial \tilde{n}} \frac{1}{R_B} dS_1 = 2\pi\phi \quad (1.119)$$

where \iint^* denotes the finite-part integral. The volume integral in Eq. (1.118) is identically zero as before (see Appendix A). Combining these results gives the supersonic boundary integral equation:

$$\phi(\mathbf{P}) = \frac{1}{2\pi} \iint_S \frac{1}{R_B} \frac{\partial \phi}{\partial \tilde{n}} - \phi \frac{\partial}{\partial \tilde{n}} \frac{1}{R_B} dS \quad (1.120)$$

where finite-part integration is implied where needed.

1.3.3 Unified Boundary Integral Equation

The above boundary integral equations for sub- and supersonic flow are very similar. For convenience, the two are combined here by introducing a factor κ . Let

$$\kappa = \begin{cases} 4\pi, & M_\infty < 1 \\ 2\pi, & M_\infty > 1 \end{cases} \quad (1.121)$$

Recalling that R_β and R_B are the same (the difference in notation was introduced only to distinguish the two flow cases), Eqs. (1.114) and (1.120) may both be expressed by

$$\phi(\mathbf{P}) = \frac{1}{\kappa} \iint_{S \cap D_P} \frac{1}{R_\beta} \frac{\partial \phi}{\partial \tilde{n}} - \phi \frac{\partial}{\partial \tilde{n}} \left(\frac{1}{R_\beta} \right) dS \quad (1.122)$$

where $S \cap D_P$ is the intersection of S and the domain of dependence of point P and finite-part integration is implied where needed. For supersonic flows, $S \cap D_P$ is the same as the region S shown in Fig. 1.2. For subsonic flows, this is the entire configuration surface, as D_P is everywhere.

An alternate derivation of the unified boundary integral equation is given in Appendix B. The derivation there relies more directly on Green's functions as opposed to Green's identities.

Equation (1.122) shows that the velocity potential ϕ may be determined at any point in the flowfield based on the distribution of ϕ and $\frac{\partial \phi}{\partial \tilde{n}}$ on S . The perturbation potential, ϕ , given by Eq. (1.122) will be a solution to the Prandtl-Glauert equation.

The function $1/R_\beta$ is the Green's function for the Prandtl-Glauert equation [25, 44]. For convenience, let

$$G = \frac{1}{R_\beta} \quad (1.123)$$

The surface contributions in Eq. (1.122) may be recognized as perturbations due to distributions of sources and doublets on the body surface [43]. In order to use doublet and source sheets to model a body inserted into the freestream, a fictitious *inner flow* must

be defined inside S . This is because doublet and source sheets represent relative jump discontinuities in the velocity potential and its normal derivative, respectively, rather than their actual values [45, 46]. Let this inner flow be denoted by its perturbation potential ϕ_i . Because all points of interest are external to S or lie on the external surface of S , they are unaffected by ϕ_i . Thus, including this inner potential in Eq. (1.122) gives

$$\kappa\phi(\mathbf{r}) = \iint_S G \left(\frac{\partial\phi}{\partial\tilde{n}} - \frac{\partial\phi_i}{\partial\tilde{n}} \right) dS - \iint_S (\phi - \phi_i) \frac{\partial G}{\partial\tilde{n}} dS \quad (1.124)$$

The term $G \left(\frac{\partial\phi}{\partial\tilde{n}} - \frac{\partial\phi_i}{\partial\tilde{n}} \right)$ is the contribution of a compressible source sheet that induces a jump in $\frac{\partial\phi}{\partial\tilde{n}}$ equal to $\sigma = \frac{\partial\phi}{\partial\tilde{n}} - \frac{\partial\phi_i}{\partial\tilde{n}}$, and $(\phi - \phi_i) \frac{\partial G}{\partial\tilde{n}}$ is the contribution of a compressible doublet sheet that induces a jump in ϕ equal to $\mu = \phi - \phi_i$. The functions G and $\frac{\partial G}{\partial\tilde{n}}$ are known as the source and doublet *kernel functions*, respectively.

Substituting in the definitions of σ and μ , Eq. (1.124) becomes

$$\kappa\phi(\mathbf{r}) = \iint_S G\sigma - \mu \frac{\partial G}{\partial\tilde{n}} dS \quad (1.125)$$

Equation (1.125) is referred to as the boundary integral equation for compressible, inviscid flows [47] and is the basic equation upon which aerodynamic panel methods are based.

In Eq. (1.125), the doublet influence on the velocity potential has the opposite sign from that of the source influence. Because of this, panel methods traditionally define a surface distribution of doublets as inducing flow through the surface in the direction opposite the normal vector (e.g. see [33, 45]). This is different from the standard definition of a doublet where flow is induced in the same direction as the normal vector (c.f. Eq. (6.9) in [48]). For consistency with the literature, the traditional panel method definition is here maintained.

One other matter of convention needs to be discussed. In the derivation of the boundary integral equation, the surface normal vector, \mathbf{n} was defined as pointing out from the flow region (i.e. into the body). This is standard in the literature regarding potential flow and Green's identities (e.g. see [2, 21, 43, 46]). However, once the focus shifts from fundamental mathematics to implementation of a panel method, the convention changes as well, and the

normal vector is defined as pointing into the flow region (i.e. out of the volume enclosed by S) [2]. This has two consequences. Since the source strength is defined as $\sigma = \frac{\partial\phi}{\partial\tilde{n}} - \frac{\partial\phi_i}{\partial\tilde{n}}$, any terms containing σ change sign. The doublet kernel function, $\frac{\partial G}{\partial\tilde{n}}$, also changes sign due to $\tilde{\mathbf{n}}$ in the denominator. Thus, when the surface normal vector is defined as pointing into the flow, Eq. (1.125) becomes

$$\kappa\phi(\mathbf{r}) = \iint_S -G\sigma + \mu\frac{\partial G}{\partial\tilde{n}}dS \quad (1.126)$$

This shift in notation will now be applied, and all subsequent equations are given assuming \mathbf{n} points into the flow.

1.4 Panel Methods

Equation (1.126) provides an elegant description of the flow field based on distributions of sources and doublets on the boundary of the flow field. It will now be shown how this equation may be used to specify a previously-unknown flowfield based on a given geometry and flow condition. Methods for doing so are called panel methods.

1.4.1 Solving the Boundary Integral Equation

From Eq. (1.126), it can be seen that finding the velocity potential about a given body for a given freestream condition amounts to determining the distribution of source and doublet strength across the surface of the body. It is not currently possible to determine these exactly for a general body. Thus, an approximate solution is sought. If S is comprised of N discrete surfaces (referred to as panels), then Eq. (1.126) may be written as

$$\kappa\phi(\mathbf{r}) = \sum_{i=1}^N \left[\iint_{S_i} -G\sigma_i + \mu_i\frac{\partial G}{\partial\tilde{n}}dS_i \right] \quad (1.127)$$

where σ_i and μ_i are distributions of source and doublet strength defined across the surface of each panel.

To model lifting flows, there may also be a wake surface trailing behind lifting surfaces. Wakes are modeled using doublet-only panels [49]. The doublet strength on the wake panels

is equal to the jump in potential from the lower to upper surfaces of the wake sheet (i.e. on the wake $\mu = \phi_u - \phi_l$ [46]). For N_w wake panels, Eq. (1.127) then becomes

$$\kappa\phi(\mathbf{r}) = \sum_{i=1}^N \left[\iint_{S_i} -G\sigma_i + \mu_i \frac{\partial G}{\partial \tilde{n}} dS_i \right] + \sum_{j=1}^{N_w} \iint_{S_{w_j}} \mu_j \frac{\partial G}{\partial \tilde{n}} dS_{w_j} \quad (1.128)$$

Thus, the potential at any point in the flow field may be determined as the sum of source and doublet influences integrated across each panel. The gradient of this may also be taken to find the velocity induced at any point in the flow field.

In addition to approximating the surface by a discrete set of panels, the source and doublet strengths are assumed to follow some analytic distribution over each panel (i.e. constant, linearly-varying, etc.). Most often, panel methods incorporate a linear distribution of doublet strength and a constant distribution of source strength across each panel, or quadratic and linear distributions of source and doublet strength, respectively. This is because the source strength represents a jump in velocity across a panel, while the gradient of doublet strength represents a jump in velocity. Some panel methods instead use constant distributions for both sources and doublets; however, this is typically only done for subsonic methods.

For a given body and freestream condition, the values of μ and σ on each panel are not known a priori. These distributions may be written in terms of some finite set of unknown source and doublet parameters. By enforcing as many boundary conditions as there are unknown source and doublet parameters, a linear system of equations may be formed. This system of equations will be linear because the induced potentials and velocities are linear in the source and doublet strengths. Then, by solving this system of equations, the unknown source and doublet parameters may be determined. Once these are known, the flow properties at any point on the body or in the flow field are easily obtained.

1.4.2 Differences Between Subsonic and Supersonic Flows

The Prandtl-Glauert equation is equally valid for both subsonic and supersonic flows. As such, panel methods have been successfully applied to both types of flows. While many similarities exist between modeling these two types of flows using panel methods, there are some important differences.

The primary difference between subsonic and supersonic flow is that the Prandtl-Glauert equation is elliptic when $M_\infty < 1$ and hyperbolic when $M_\infty > 1$. As discussed previously, this means that disturbances introduced in supersonic flow only propagate within a limited region, rather than throughout the flow field, as with subsonic flows. With a subsonic panel method, every panel exerts an influence on every part of the flow. With a supersonic panel method, this is not the case. Hence, domains of dependence (described previously) must be taken into consideration in supersonic flows.

These domains of dependence also relate to whether individual panels in a supersonic panel method are sub- or superinclined. As discussed previously, surfaces may be subinclined, Mach-inclined, or superinclined (see Section 1.2.3). Mach-inclined panels are not allowed in panel methods as the local-scaled coordinate transformation (see Section 3.2) breaks down for such panels [2, 33]. Examples of the other two types of panels are shown in Fig. 1.1. For a superinclined panel, no point on its surface can exert an influence on any other point on its surface, since the entire panel falls outside of the domain of influence of any point on the panel. This is not the case with subinclined panels for which points on their surface can influence other points on the panel.

Additionally, a supersonic panel method requires continuity of doublet strength between panels, while a subsonic method does not [2, 50]. This is because a discontinuity in doublet strength at any point on the surface results in a line vortex. These do not cause issues in subsonic flow because the effect of these vortices die off with distance from the vortex. However, in supersonic flow, the infinite induced velocities at the vortex core also exist on the Mach cone extending from the vortex. These singularities cause serious numerical issues in supersonic panel methods [50].

1.4.3 History

Panel methods belong to a class of numerical methods called boundary element methods (BEM). Such methods employ the method of Green's functions using discretized boundary meshes as described above [44,51]. Solving aerodynamic problems using a boundary integral equation (such as Eq. (1.122)) likely began with Prandtl in 1921 in connection with lifting-line theory [47]. Numerical boundary element methods began to appear in the 1950's and 60's, shortly after the advent of modern computers [52].

In the development of panel methods, certain simplified precursors to the true panel method appear. A well-known example of this is the vortex lattice method. Vortex lattice method is essentially similar to a panel method, except that lifting surfaces are assumed to be thin, so there is no inner potential. Because of this thin-surface assumption, sources, if present, are used to approximate the effect of thickness [30,46]. Certain other methods may be obtained based on various assumptions regarding the thickness or slenderness of the configuration being analyzed. However, the focus of this work is on true panel methods, which make no such limiting assumptions and so model the configuration exactly.

An in-depth discussion of early potential flow numerical panel methods was given in 1967 by Hess and Smith [53]. In 1984, Miranda [54] reviewed computational aerodynamics with a focus on linear, three-dimensional panel methods. An excellent review of supersonic panel methods was presented in 1990 by Erickson [50]. Erickson discussed five supersonic panel methods then extant: the Woodward-Carmichael code [55,56], USSAERO (also called Woodward-II) [57–59], PAN AIR [2,33,60], HISSS [61], and QUADPAN [62,63]. Erickson also reviewed SOUSSA [64], which was developed to handle supersonic flows but altogether failed to do so [50].

Most of these legacy codes are epitomized by PAN AIR, which may be regarded as the most advanced. Though it came later, the developers of HISSS explicitly state that HISSS is an imitation of PAN AIR [61]. The exception to this rule is QUADPAN, developed by the Lockheed company. QUADPAN was originally developed as a low-order, subsonic code, and was later expanded to model supersonic flows [50,54,62,63]. Of all these methods, PAN

AIR is still in use by aircraft designers and researchers (e.g. see [14, 18, 19]). This is mostly due to the fact that only the source code for PAN AIR is publicly available [65]. Most other methods were developed for in-house use, and so only technical reports and academic papers describing them are available.

These legacy codes have several attributes in common. For one, they are all based on having the computational mesh represented by a structured grid. A structured grid is stored as a set (or sets) of points arranged in rows and columns. Each set of four neighboring points defines a panel. These codes also tend to use the Morino boundary condition formulation, the derivation of which is given in Chapter 2. These methods also generally employ quadratic-doublet-linear-source panels. QUADPAN again is an exception to this as it employs linear-doublet-constant-source panels [50].

Several other supersonic panel methods were developed in the 1980's and 90's but were not mentioned by Erickson. The most notable is MARCAP, developed around 1988 at Mitsubishi [66, 67]. Like its predecessors, MARCAP used structured meshes and employed the Morino formulation. Unlike PAN AIR and HISS, MARCAP used linear doublet and constant source panels. Despite this simplification, results obtained from MARCAP compared favorably with those obtained from PAN AIR [7, 67, 68]. Other supersonic panel methods include NLRAERO [69] and the British Aerospace Supersonic Panel Program [70], both based heavily on the Woodward/USSAERO codes [71].

Since Erickson's review, several more modern panel methods have been developed. In addition to supersonic methods, unstructured, incompressible/compressible-subsonic panel methods have experienced a boom in recent years (see for example [72–78]). The basic formulations for such methods are well-described in Katz and Plotkin's book, *Low-Speed Aerodynamics* [46]. The vast majority of such codes have also been developed for in-house use and so are not publicly available. An exhaustive survey of such methods is not justified here, but some of the more notable features will be highlighted.

One of the most notable aspects of these modern incompressible methods has been the increased use of constant-doublet (vortex ring) panels [72, 77], compared to the higher-

order distributions used in legacy incompressible methods, such as that of Johnson [45, 79]. Maskew [80] has shown how using constant-strength doublet panels produces results comparable in accuracy to quadratic doublet panels for the same control point density. This is advantageous because the calculation of constant-doublet influences is computationally cheaper than higher-order formulations [54, 72, 80].

Also notable is that unstructured meshes have been universally adopted among these incompressible methods. Unlike structured meshes, which store the mesh vertices in a grid, unstructured meshes store vertices and panels simply as lists of polygons in space with the arrangement of and relationships between panels not specified. Along with unstructured meshes, efficient methods have been developed for automatically generating [72] and relaxing wakes [46, 74, 77].

Beyond these modernized incompressible methods, two unstructured, supersonic panel methods have appeared since the time of Erickson’s review: ZONAIR and CPanel. The most notable of these two is ZONAIR [81, 82] produced by ZONA Technologies, which has become a common alternative to PAN AIR for aerodynamic design and analysis (e.g. see [12, 16]). However, its use as presented in the academic literature is limited, and being a proprietary code, very little is published regarding its implementation.

The most recent attempt at a modern supersonic panel method is that of Davis, a graduate student at CalPoly, in 2019. Davis provided an update to the in-house code CPanel that allowed it to model supersonic flows over bodies consisting only of subinclined panels (superinclined panels are not allowed) [7, 68]. The supersonic implementation in CPanel is mostly inspired by PAN AIR [2] and MARCAP [66]. Davis used the influence coefficient calculation method presented by Ehlers et al. [2]. With this update, Davis reports that CPanel provides predictions for supersonic flows which favorably compare with results obtained from PAN AIR [33] and MARCAP [66] for a variety of configurations [7]. Davis also provides an efficient and intuitive method for enforcing doublet strength continuity on an unstructured mesh. However, Davis highlights several shortcomings to his method,

some of them being the inability to model wakes, difficulty modeling sharp subsonic leading edges, and poor predictions at wing-body junctions.

PAN AIR remains probably the most-used panel method for supersonic flow analysis (e.g. see [10,14,17–19,83]). This is likely due to it being freely available and reasonably-well documented [2,33,60]. It has also been extensively evaluated and vetted in the years since its development.

1.5 Existing Difficulties and Weaknesses

Apart from the appearance of ZONAIR and CPanel, the development of supersonic panel methods appears to have been stagnant (based on the available literature) for the past 35 years. As such, several improvements have been made to panel methods in general that have yet to be applied to supersonic panel methods. In addition, there exist outstanding weaknesses to supersonic panel methods and gaps in understanding their theory. The following difficulties and weaknesses have been identified:

1.5.1 Lack of clarity regarding the need for higher-order singularity distributions

It is widely accepted that when modeling supersonic flows, the distribution of doublet strength across the configuration surface must be continuous [2,50,50]. If the distribution is not continuous, then line vortices appear at boundaries where the doublet strength is discontinuous. As discussed previously, many subsonic panel methods use constant-strength doublet panels, which results in a discontinuous distribution of doublet strength across the configuration. However, this does not pose a problem for subsonic methods because the vortex singularity quickly dies off with distance from the panel edge, and so solutions obtained are still well-behaved [50,80]. However, in supersonic flow, these vortex singularities also exist along the Mach cone boundaries and can wreak numerical havoc on unlimited points downstream [50]. For this reason, panels having at least a linear distribution of doublet strength are used universally among supersonic panel methods [7,33,61,66,82]. Such

higher-order distributions enforce continuity of doublet strength along panel edges, and so no line vortices are created.

Doublet distributions that enforce continuity may be either linear or quadratic; distributions of even higher degree are possible, but no examples of such methods are to be found. Most legacy codes employed quadratic doublet distributions [33, 61]. Besides the need for a continuous distribution of doublet strength, several other arguments have been made in favor of quadratic distributions over linear distributions. One of these comes from the fact that panel methods are often paired with three-dimensional boundary-layer simulation codes to apply viscous corrections to the inviscid results. It has been reported that these boundary-layer simulations respond very poorly to errors introduced by using linear doublet distributions instead of quadratic doublet distributions [61, 84].

It has also been claimed that quadratic distributions provide greater accuracy. In developing a supersonic lifting-surface method, Mercer et al. [85] argue that quadratic doublet distributions are also necessary in order to obtain accurate estimates of lateral stability derivatives for configurations with small dihedral. They base this argument off of a paper by Rubbert [86]. In this paper, Rubbert analyzes various contributions to the perturbation velocity potential. He shows that for configurations with small dihedral, the dominant side force effects appear as second-order terms in sideslip, angle of attack, etc. Because of this, Rubbert states, “the present second-order theory must be accompanied by a numerical method capable of producing accurate velocity gradients in the first-order solution.” [86]. Linear doublet distributions result in a constant distribution of velocity, whereas quadratic doublet distributions result in a linear distribution of velocity [45, 85]. Thus, quadratic distributions have been supposed to be necessary for modeling certain aerodynamic effects, such as sideforce due to sideslip.

Despite these arguments, there has been little direct evaluation of the effect of linear versus quadratic doublet distributions on solutions obtained from a panel method. Codes based on linear doublet distributions (such as MARCAP and CPanel) have been shown to compare favorably with higher-order codes, such as PAN AIR. However, no supersonic

panel method has yet been developed that implements both orders of distributions, making a direct comparison difficult.

Often discussed in the literature is the assertion that a panel method is not truly high-order unless it accounts for higher-order panel geometries (i.e. curved panels) [2,45,87]. Hess et al. report that studies have shown that higher-order methods which fail to account for panel geometry do not show more rapid convergence with mesh refinement than low-order methods [87]. However, no examples of supersonic methods using higher-order singularity distributions and curved panels are to be found due to mathematical difficulties in deriving the influence coefficients for such [2,33].

As a final note, it has been demonstrated through the development of the supersonic vortex-lattice code VORLAX that horseshoe vortices and vortex rings may be used to simulate bodies in supersonic flow [30]. To do this, a special formulation of the influence coefficients is used which avoids Mach-line singularities. Other techniques for avoiding Mach-line singularities induced by supersonic vortex filaments have also been put forward [88]. However, the discussion of such constant-strength methods for supersonic flow is beyond the scope of this work.

1.5.2 Lack of clarity regarding trade-offs between boundary-condition formulations

To solve Eq. (1.126), boundary conditions must be applied. While there are multiple ways to enforce the zero-normal-flow boundary condition which are mathematically equivalent (see Chapter 2), it has been noted that, in practice, different formulations can give differing results. For example, Tinoco and Rubbert [60] state that the Morino formulation (which uses both sources and doublets) “is only slightly more expensive computationally and is much more insensitive to paneling” compared to formulations which do not include sources. A broad review of the literature has revealed that the Morino formulation is most commonly used after first being popularized by the panel method SOUSSA [50,64], which was developed by Luigi Morino. However, Erickson points out that the Morino formulation typically introduces some error as it assumes the internal perturbation potential is every-

where exactly zero, which is rarely the case [50]. Other boundary condition formulations have been presented, either Neumann or Dirichlet, some using only sources, some using only doublets, and others using both. Direct studies into the relative advantages and disadvantages of these formulations have been limited. In addition, no attempts to model supersonic flows without using sources could be found in the literature. Thus, there is significant room for improving boundary condition formulations for supersonic flow.

1.5.3 Use of structured meshes

Computational methods for years employed structured meshes to represent geometries. This is because computer hardware limited the amount of memory which could be used to store the mesh. Structured meshes represent a surface geometry as a rectangular grid of points with each set of neighboring points defining a four-sided panel. Since the exact structure of the grid is known, the surface is entirely defined only by the corner points. While computationally inexpensive to store and process, structured meshes are difficult to generate and refine. They also force undesirable clustering of panels for certain types of geometry [72]. Rather than being a fundamental limitation of panel methods, the use of structured meshes is simply an implementation issue.

As computing power has increased over time, numerical methods have shifted to using unstructured meshes, which are simply collections of triangular surface facets defined independently of one another. Unstructured meshes are more memory-intensive, but they are much easier to generate and refine. Among existing supersonic panel methods, PAN AIR uses structured meshes [33] and CPanel (which is limited in applicability) uses unstructured meshes [7]. ZONAIR can use a mixture of both [82]. Aside from the basic implementation of unstructured meshes, which has been demonstrated with supersonic panel methods, there may be opportunities for improvement. In particular, no supersonic panel method has yet been developed that implements quadratic doublet distributions on an unstructured mesh.

1.5.4 Unreliable treatment of superinclined panels

As discussed previously, superinclined panels are panels inclined to the freestream at an angle greater than the Mach angle (i.e. they lie outside of their own domains of influence). Due to the hyperbolic character of the Prandtl-Glauert equation, these panels cannot represent solid surfaces [2, 50]. However, they are useful for closing engine inlets and outlets and similar features [89–92]. In doing so, they serve the purpose of cancelling out the flow incident on their upstream surface.

Many users of supersonic panel methods report being unable to obtain reliable results when superinclined panels are included (see for example [14, 18, 19, 93]). The developers of ZONAIR recommend superinclined panels not be used [82]. One method which has been proposed for circumventing this issue entirely is an equivalent Mach number transformation [16, 94]. However, the mathematical validity of this method is dubious. Hence, a more in-depth study into the mathematics and use of superinclined panels is needed.

1.5.5 Minimal use of efficient matrix algorithms

To obtain accurate flow solutions about complex configurations, surface meshes with a large number of panels are often required. This in turn results in a large system of equations to be solved. Thankfully, the system of equations is linear. Direct matrix methods, such as LU decomposition, have $O(N^3)$ complexity, which for such a large system means a great deal of computational effort is spent solving the linear system. All other operations of a linear panel method (calculating influence coefficients, locating adjacent panels, determining domains of dependence, etc.) are $O(N^2)$ operations, making the matrix solver the most expensive part of the overall computation for large meshes.

The efficient solution of the system of equations arising from an elliptic BEM has been investigated previously (e.g. see [95–104]). However, the system of equations in a supersonic panel method has not yet been considered. Most supersonic panel methods have used direct matrix methods. PAN AIR simply uses LU decomposition for solving the linear system [33].

As discussed previously, the supersonic Prandtl-Glauert equation admits limited domains of dependence and influence for any point in the flow. Because of this limited influence, the system matrix contains many elements that are identically zero. It may be possible to exploit this structure to more efficiently solve the linear system of equations.

1.5.6 Appearance of expansive shocks

As with all compressible, inviscid methods, panel methods will produce solutions in supersonic flow that include expansive shocks, which are nonphysical [21,34]. This is because expansive shocks are suppressed in reality through viscosity. Higher-fidelity methods may be formulated such that expansive shocks are suppressed (e.g. using numerical viscosity, selective upwinding, etc.). However, no such options exist for linear panel methods.

1.5.7 Constant characteristic angles

One major assumption built into the Prandtl-Glauert equation is that the Mach number of the flow is everywhere equal to the freestream Mach number. This means that the characteristic surfaces (which represent shocks and expansion fans) predicted by the Prandtl-Glauert equation are all at the same angle to the freestream. This is a reasonable approximation when the real shocks are weak and the desired flow results are close to the body. However, this approximation becomes poor for stronger shocks, non-slender bodies, and flow results far away from the body.

1.6 Alternative Aerodynamic Methods

Given the weaknesses and difficulties described in the previous section, it is appropriate to consider whether linear panel methods are the best approach to take in developing a modern computational tool for preliminary aerodynamic design and analysis. Several other options exist. A handful of methods are briefly reviewed here to justify the linear panel approach taken in the current research effort.

1.6.1 Field-Panel Methods

One class of methods closely related to linear panel methods are field-panel methods. In deriving Eq. (1.122), it is possible to use the full potential equation (Eq. (1.48)), or even the transonic small disturbance equation [31], instead of the Prandtl-Glauert equation. In doing so, the final equation is very similar to Eq. (1.122), except for the addition of a volume integral that brings in the influence of $\nabla^2\phi$ over the fluid volume, which is nonzero for these higher-fidelity models [25, 27, 32]. The method for solving this equation is very similar to a linear panel method except that an iterative scheme is required, since the volume integral term is dependent upon the calculated solution. Examples of such methods may be found in [25, 27, 32, 38, 105–107].

For field panel methods, the fluid volume surrounding the configuration must also be discretized in addition to the configuration surface. One of the major advantages of panel methods is that only a two-dimensional mesh is necessary for solving three-dimensional problems, and this advantage is lost when moving to field-panel methods. However, the volume meshing requirements for a field-panel method are not nearly as strict or extensive as for finite-volume methods. The volume mesh for a field-panel method may be relatively coarse and does not need to extend to where all perturbations from the body vanish.

Probably the most well-known example of a field panel method is TRANAIR, developed at the Boeing Company. Early versions of the code TRANAIR used a field-panel formulation to solve transonic flow problems [50, 107, 108]. However, TRANAIR later shifted to a finite-volume full potential formulation [109, 110], which will be discussed subsequently.

There is a special type of field panel method, typified by the GENESIS scheme proposed by Hunt, Plybon, and Adamson [40, 78, 111–114]. This method uses approximate basis functions to collapse the volume terms present in field panel formulations back onto the configuration surface. Thus, the method again becomes purely a boundary element method. It was also reported that the GENESIS scheme could model rotational flows. However, no successful implementation of this scheme has ever been reported, and so its accuracy, robustness, and versatility cannot be evaluated. In 2010, Ramos [115] presented a

formulation similar to the GENESIS scheme. Ramos’s scheme applies the dual-reciprocity method to full potential flows. Despite reporting good results, Ramos’ method has a free parameter that requires tuning based on a knowledge of the true solution, limiting its applicability to novel situations.

Field-panel methods can capture some nonlinear flow phenomena that linear panel methods cannot. They may also be used to reliably model transonic flows and can be corrected to suppress expansive shocks. However, the added computational cost of meshing the fluid volume, calculating field influences, and iteratively solving the nonlinear system is significant.

1.6.2 Full Potential Methods

Another class of related methods are full potential methods. Examples of full potential methods are described in [26, 54, 109, 110, 116]. These methods are many and varied, and so only a brief overview is given here. Like field-panel methods, full potential methods also use the full potential equation (Eq. (1.48)), but solve it using finite-volume or finite-element techniques, rather than the method of Green’s functions. Through use of Cartesian grids, and by solving the potential equation (which only has one unknown), these methods tend to be much faster than traditional finite-volume methods (such as CFD and Euler methods). However, the potential formulation suffers the same weakness as was described for panel methods, namely, that flow through a shock wave is not isentropic and so will not remain irrotational [21], making such potential-based methods only approximate for supersonic flows. Thus, the use of such methods is typically restricted to transonic and low supersonic speeds, where shocks are weak [26]. Full potential methods, being inviscid, also produce expansive shocks in solutions unless artificial viscosity is added to the formulation [110].

In reviewing such methods in 2020, Crovato et al. [116] state that such full potential methods are still too computationally expensive to be practical for preliminary design and analysis. While still less computationally expensive than Euler and Navier-Stokes methods due to there only being one unknown variable, they are much more expensive than linear

panel methods. Their usefulness comes from the ability to model nonlinear flow phenomena with greater accuracy than panel methods.

1.6.3 Euler and Navier-Stokes Methods

The highest levels of fidelity in aerodynamic analysis come from using finite-volume solvers based on the full Euler or Navier Stokes equations (Reynolds-averaged or not). These are well-known, and examples are easily found. While capturing more of the flow physics and being more versatile than the methods already discussed, these methods are also very computationally expensive. They are valuable for late-stage design and analysis. Much of the research effort into computational fluid dynamics in recent years has been directed at these methods. Because of this, they are highly optimized and robust. However, their computational cost is still significant.

1.7 Research Objectives

After reviewing the relevant literature, it is clear that a modern investigation into linear panel methods is needed. Compared to the other methods reviewed above, linear panel methods offer several advantages, particularly speed and ease of use. Several weaknesses of panel methods have been identified which may be alleviated through focused investigation.

The primary objective of this research effort is to **identify the best approach to modeling supersonic flows using a linearized panel method on unstructured meshes**. As described in the introduction, linear panel methods are not a new concept, even for modeling supersonic flows. However, their theoretical development has been essentially stagnant for several decades. There is need for a thorough investigation into best practices for these methods given advancements made in computational science in recent years.

In approaching this primary objective, the following four secondary objectives must be met:

1. Characterize the impact of varying singularity distributions on the accuracy, speed, and robustness of an unstructured panel method.

2. Characterize the impact of boundary condition formulation on the accuracy, speed, and robustness of an unstructured panel method.
3. Assess the performance of various matrix solvers for the unique matrix equation appearing in supersonic panel methods.
4. Identify a viable path forward for the development of a recursive domain of dependence search algorithm and assess its impact on solution speed.

1.7.1 Objective 1: Characterize the impact of varying singularity distributions on the accuracy, speed, and robustness of an unstructured panel method.

As stated previously, there has not yet been a direct comparison of varying singularity distributions for an unstructured, supersonic panel method. There also has yet to be an implementation of quadratic doublets and linear sources on an unstructured mesh. As part of this work, both higher-order (quadratic doublets and linear sources) and lower-order (linear doublets and constant sources) distributions will be implemented as part of an unstructured panel method. The relative behavior of each distribution will be characterized. Doing so will give a clear picture of the trade-offs between these two options and clarify best practices for modern, unstructured, supersonic panel methods.

1.7.2 Objective 2: Characterize the impact of boundary condition formulation on the accuracy, speed, and robustness of an unstructured panel method.

Unlike varying singularity distributions, others have reported comparisons between varying boundary condition formulations. Several claims have been put forward as to the relative merits of each, particularly for supersonic flows. However, this aspect of supersonic panel methods has yet to be explored on unstructured meshes. Since many of the differences between the performance of the different boundary condition formulations have been attributed to panelling effects, it is anticipated that the use of unstructured meshes has the potential to alter the current consensus regarding boundary condition formulations. In

addition, the development of a novel, source-free, Dirichlet boundary condition formulation will be presented.

1.7.3 Objective 3: Assess the performance of various matrix solvers for the unique matrix equation appearing in supersonic panel methods.

As discussed previously, no novel matrix solution method has yet been applied specifically to solving the unique linear system which appears for supersonic flows. Knowing that a large portion of the system matrix is empty should lead to more efficient matrix solution methods than traditional dense matrix solvers. A wide range of matrix solvers will be investigated and their performance for the supersonic case characterized. Novel approaches to solving the supersonic system of equations will also be investigated.

1.7.4 Objective 4: Identify a viable path forward for the development of a recursive domain of dependence search algorithm and assess its impact on solution speed.

In order to save time when calculating influence coefficients, a check is first made to determine whether the panel lies in the domain of dependence of the current control point. As described in [117], it is possible to determine some of the information necessary for this domain of dependence check recursively, thus reducing the overall computation time. This is because if point A belongs to the domain of dependence of point B , then all points in the domain of dependence of point A will also belong to the domain of dependence of point B . As part of this research effort, a method for determining all domains of dependence for the mesh using this recursive principle will be refined and implemented. Once implemented, its effect on the performance of the method will be characterized.

1.8 Pilot Code

In addition to the objectives listed above, the purpose of this work is to provide a reasonably complete exposition of the theory and implementation of compressible panel methods on unstructured meshes. As such, much of this work is devoted to gathering, condensing, and clarifying existing research. The panel method as described in this work has been implemented in an open-source pilot code. This code is called MachLine (Multi-order Approach to Calculating High-speed Linear aerodynamics) and is freely available online [118, 119]. A brief description of MachLine is given in Appendix C. All results presented in this work were obtained using MachLine.

1.9 Current Scope

The rest of this work proceeds as follows. Chapter 2 derives the boundary condition formulations examined here and describes how they may be used to determine the unknown source and doublet strengths. Chapter 3 deals with basic elements of a panel method for parsing and analyzing unstructured meshes. Chapter 4 describes how domains of dependence are calculated, including a novel recursive domain of dependence algorithm. Chapter 5 discusses the doublet and source distributions on each panel, giving the formulas necessary for determining the distribution parameters and the influence of a panel on a given point in space. Chapter 6 presents the investigation into matrix solution methods for the supersonic linear system of equations. Chapter 7 describes how relevant flow properties are calculated once the source and doublet distributions have been determined. Chapter 8 presents investigations into various numerical characteristics of the pilot code, such as grid convergence and sensitivity to paneling. Chapter 9 presents an evaluation of the accuracy of the panel method as implemented in the pilot code compared against analytic flow solutions and experimental data. Chapter 10 summarizes the research effort and provides recommendations for future work.

CHAPTER 2

BOUNDARY CONDITIONS

Newcomers to panel methods often find the idea of an internal flow field to be strange in that no such flow exists inside a real wing. However, it must be remembered that we are using sources, doublets, and boundary conditions to create the flow fields, and flow will exist on either side of the source-doublet panels. - Larry Erickson [50]

Well, what are your boundary conditions? - Dr. Thom Fronk

The discretized BIE, Eq. (1.128), is solved by applying boundary conditions. There are a variety of methods for applying boundary conditions to the BIE. These may be broadly categorized as Dirichlet (potential-based) or Neumann (velocity-based), though the two may also be mixed. This work focuses on Dirichlet formulations, though Neumann formulations will be discussed briefly.

2.1 Dirichlet Boundary Condition Formulations

For impermeable surfaces, there should be no mass flux through the surface. This may be expressed as [2]

$$\rho \mathbf{u} \cdot \mathbf{n} = 0 \tag{2.1}$$

where ρ is the fluid density, \mathbf{u} is the fluid velocity, and \mathbf{n} is the normal vector to the surface. Note that boundary conditions for permeable surfaces with prescribed mass flux are possible. In such cases, the boundary condition is

$$\rho \mathbf{u} \cdot \mathbf{n} = \dot{m}_{\text{des}} \tag{2.2}$$

where \dot{m}_{des} is the prescribed mass flux through the surface. Implementation of such boundary conditions is described in the literature (e.g. see [2, 120]). However, this work focuses purely on the zero-mass-flux condition.

Recall the *total velocity potential* is defined as

$$\Phi = U(\phi_{\infty} + \phi) \quad (2.3)$$

where $\mathbf{u} = \nabla\Phi$, U is the magnitude of the freestream velocity, ϕ_{∞} is the freestream potential, and ϕ is the perturbation potential.

For the sake of introduction, first consider an incompressible flow. For an incompressible flow, the density is everywhere equal to the freestream density, and Eq. (2.1) becomes

$$\rho_{\infty}(\mathbf{U} + \mathbf{u}') \cdot \mathbf{n} = 0 \quad (2.4)$$

where $\mathbf{U} = U\nabla\phi_{\infty}$ is the freestream velocity, $\mathbf{u}' = U\nabla\phi$ is the perturbation velocity, and ρ_{∞} is the freestream density. Dividing out the density and applying irrotationality yields

$$U\nabla\phi \cdot \mathbf{n} = -\mathbf{U} \cdot \mathbf{n}$$

This then becomes [45]

$$\frac{\partial\phi}{\partial n} = -\hat{\mathbf{c}} \cdot \mathbf{n} \quad (2.5)$$

where $\hat{\mathbf{c}} = \frac{\mathbf{U}}{U}$ is the direction of the freestream flow. Equation (2.5) must be satisfied on the outer surface of the configuration in an incompressible flow in order for there to be zero mass flux through the configuration surface.

Now consider a compressible flow. As shown in Section 1.1.8 (see Eq. (1.85)), the perturbation mass flux is (to a linear approximation) equal to [2, 21]

$$\rho\mathbf{u} - \rho_{\infty}\mathbf{U} \approx \rho_{\infty}\mathbf{w} \quad (2.6)$$

where \mathbf{w} is the reduced current velocity discussed previously (see Eq. (1.72)). Using this, for compressible flows, the linearized mass flux boundary condition is given by

$$(\rho_\infty \mathbf{U} + \rho_\infty \mathbf{w}) \cdot \mathbf{n} = 0 \quad (2.7)$$

Note that, because of the definition of \mathbf{w} , this approximation becomes exact for incompressible flows.

Equation (2.7) may be rearranged to yield

$$\mathbf{w} \cdot \mathbf{n} = -\mathbf{U} \cdot \mathbf{n} \quad (2.8)$$

Applying the definition of the reduced current velocity (Eq. (1.72)) and dividing out U , this becomes

$$\mathbf{n}^T [B] \nabla \phi = -\hat{\mathbf{c}}^T \mathbf{n} \quad (2.9)$$

Because $\tilde{\mathbf{n}} = [B] \mathbf{n}$ and $[B] = [B]^T$ [33], this may be rearranged to yield

$$\frac{\partial \phi}{\partial \tilde{n}} = -\hat{\mathbf{c}} \cdot \mathbf{n} \quad (2.10)$$

Equation (2.10) must be satisfied on the outer surface of the body under consideration in order for the linearized zero-mass-flux boundary condition to hold.

While convenient for use with the BIE (Eq. (1.122)), the linearized zero-mass-flux boundary condition and Dirichlet boundary conditions in general have some drawbacks. The linearized zero-mass-flux boundary condition does not result in the flow velocity being everywhere tangent to the configuration surface. This is because the linearized mass flux and the flow velocity are not parallel in most cases [21]. Enforcing flow tangency explicitly may be done using Neumann formulations (for example, see [121]), and this will give somewhat different results from the zero-mass-flux boundary condition. However, the implementation and analysis of Neumann boundary conditions is beyond the scope of this work.

Another drawback to Dirichlet boundary condition formulations is that wake-body interactions require special treatment. Since the wake sheet (being a sheet of doublets) represents a jump in velocity potential from one side of the sheet to the other, any junctions between the wake and part of the configuration being modeled must result in a discontinuity in doublet strength on the configuration surface. For example, on a traditional aircraft, the wake is shed from the main wing and trails aft along the fuselage. The surface must then have a discontinuity in doublet strength at the line along which the wake meets the fuselage; otherwise the interaction between the wake and fuselage will not properly be captured [33]. While the theory of this is simple, its implementation is not. As now implemented, MachLine is not capable of modeling this interaction. This is an active area of research moving forward.

Despite these drawbacks, Dirichlet formulations are desirable because they tend to be faster than Neumann formulations. This is because the calculation of potential influences is computationally cheaper than calculating velocity influences. In addition, Dirichlet formulations may be derived where the source strengths can be calculated directly, reducing the size of the linear system of equations significantly.

2.1.1 Morino Formulation

By far the most common method for enforcing Eq. (2.10) among compressible panel methods is the Morino formulation. It is used in PAN AIR [2], MARCAP [66], QUADPAN [54, 122], CPanel [7, 68], and ZONAIR [82], among others.

To derive the Morino formulation, recall that the local source strength is equal to the jump in the conormal derivative of velocity potential across the surface (i.e. $\sigma = \frac{\partial \phi}{\partial \bar{n}} - \frac{\partial \phi_i}{\partial \bar{n}}$). To take advantage of this, ϕ_i is chosen to be identically zero everywhere. If $\phi_i \equiv 0$, then it must be that

$$\frac{\partial \phi_i}{\partial \bar{n}} = 0 \tag{2.11}$$

and

$$\sigma = \frac{\partial \phi}{\partial \tilde{n}} \quad (2.12)$$

Combining this with Eq. (2.10) leads to [2, 7, 66]

$$\sigma = -\hat{\mathbf{c}} \cdot \mathbf{n} \quad (2.13)$$

Using the Morino formulation, the source strengths are calculated simply from the freestream velocity and the surface geometry.

Once the source strengths are known, the doublet strengths are determined based on the condition that the inner perturbation potential needs to be zero everywhere. Note this was initially chosen almost arbitrarily. Now it will be enforced. If ϕ_d is the perturbation velocity potential induced by the distribution of doublets on S and ϕ_s is the perturbation velocity potential induced by the distribution of sources on S , then for the Morino formulation, it must be that

$$\phi_d + \phi_s = 0 \quad (2.14)$$

everywhere inside the configuration. How this may be used to determine the unknown doublet strengths will be discussed subsequently.

One advantage to the Morino formulation is that the non-zero mass flux boundary condition (Eq. (2.2)) may also be used on parts of S . For example, if a set of panels represents a velocity inlet on the surface for which the mass flux is known, then this results only in a change in the source strength on those panels. However, the implementation of this is beyond the scope of this work.

2.1.2 Source-Free Formulation for Zero Normal Mass Flux

It is possible to derive other Dirichlet boundary condition formulations that satisfy Eq. (2.10). In particular, it is possible to do so without using sources. Currently, it is common for low-order, subsonic panel methods to employ only doublet panels (e.g. see [72, 80, 123]). This reduces computational expense, because source strengths and influences never need to be calculated. However, it is often claimed in the literature that using only doublet panels gives poor results compared to using both doublet and source panels (for example, see [50, 60, 67, 121]). It has been hypothesized that this is due to the boundary conditions not being enforced exactly everywhere on S [67, 121]. Because of this, some flow is “leaked” into or out of the configuration. If source panels are present, then these may serve to “absorb” this “leaked” flow. Thus, common practice has been to also include a single source panel inside the configuration if only doublet panels are used on the surface. However, examples of this are only to be found in studies using Neumann boundary conditions (for example, see [121]). Here, a source-free formulation is presented that is a Dirichlet formulation. This is entirely novel and may not suffer the same failings as source-free Neumann formulations.

The goal is to determine an inner potential such that Eq. (2.10) is satisfied without sources on S . If there are no sources on the configuration surface, then

$$\frac{\partial \phi}{\partial \tilde{n}} = \frac{\partial \phi_i}{\partial \tilde{n}} \quad (2.15)$$

Combining the above equation with Eq. (2.10) results in

$$\frac{\partial \phi_i}{\partial \tilde{n}} = -\hat{\mathbf{c}} \cdot \mathbf{n} \quad (2.16)$$

which may be rewritten as

$$\mathbf{n}^T [B] \nabla \phi_i = -\mathbf{n}^T \hat{\mathbf{c}} \quad (2.17)$$

since $[B]$ is symmetric. This will be satisfied if (sufficient but not necessary)

$$[B]\nabla\phi_i = -\hat{\mathbf{c}} \quad (2.18)$$

which becomes

$$\nabla\phi_i = -[B]^{-1}\hat{\mathbf{c}} \quad (2.19)$$

Integrating both sides and choosing the constant of integration to be zero (for simplicity) yields

$$\phi_i = -\mathbf{r}^T[B]^{-1}\hat{\mathbf{c}} \quad (2.20)$$

If the inner potential is equal to that given in Eq. (2.20), then the linearized zero-mass-flux boundary condition will be satisfied everywhere on the configuration surface.

As with the Morino formulation, the doublet strengths need to be calculated such that Eq. (2.20) is satisfied everywhere inside the configuration. However, source strengths and source influence never need to be calculated, saving some time over the Morino formulation. Unlike the Morino formulation, non-zero mass flux boundary conditions on certain panels cannot be enforced.

2.1.3 Solving for the Unknown Doublet Strengths

Both the Morino formulation and the source-free formulation involve setting the source strengths and then solving for the doublet strengths to match some desired potential inside the configuration. The source strength calculations and desired inner potentials are summarized in Table 2.1.

Table 2.1: Source strengths and desired inner potentials for the two boundary condition formulations.

Formulation	σ	$\phi_{i_{\text{des}}}$
Morino	$-\hat{\mathbf{c}}^T \mathbf{n}$	0
Source-Free	0	$-\mathbf{r}^T[B]^{-1}\hat{\mathbf{c}}$

Initially, the source strengths are known, and the doublet strengths are unknown. The doublet strengths must be determined such that the combined source and doublet influences inside the configuration equal the desired inner potential. This inner potential condition is enforced at a finite number of control points. At each of the control points, the induced potential due to the source distribution, ϕ_s , may be calculated. It must then be that the sum of the source- and doublet-induced potentials at each control point is equal to the desired inner potential, leading to the condition

$$\phi_d = \phi_{i_{\text{des}}} - \phi_s \quad (2.21)$$

For the source-free formulation, ϕ_s is simply zero.

Forcing the inner potential to equal the desired inner potential at a finite set of control points does not mean that the inner potential condition will be satisfied exactly throughout the inside of the configuration. However, if there are enough control points and they are regularly spaced, the inner potential should be close enough to the desired potential.

As will be described in Chapter 5, ϕ_d may be written as the inner product of a vector of *aerodynamic influence coefficients* and a vector of unknown doublet strengths. These doublet strengths correspond to the strength of the doublet distribution at certain points on the mesh (in this work, the mesh vertices). If there are N control points and N unknown doublet strengths, the vectors of influence coefficients at each control point together form an $N \times N$ matrix. Grouping the right-hand sides of Eq. (2.21) into a vector, this may be written as a matrix equation:

$$[AIC]\{\mu\} = \{\phi_{i_{\text{des}}}\} - \{\phi_s\} \quad (2.22)$$

where $[AIC]$ is called the *aerodynamic influence coefficient (AIC)* matrix and contains the influence of each doublet strength on each control point, $\{\mu\}$ is the vector of unknown doublet strengths, $\{\phi_{i_{\text{des}}}\}$ is the vector of desired inner potentials at each control point, and $\{\phi_s\}$ is the vector of source-induced potentials at each control point. Equation (2.22)

may be solved using any convenient matrix solver. Chapter 6 of this work presents several matrix solvers and their performance in solving this system.

2.2 Superinclined Panels

Before moving on to discuss alternatives to the Morino and source-free boundary condition formulations, a discussion regarding superinclined panels is needed. As discussed in Section 1.2.3, surfaces in supersonic flow may be subinclined, Mach-inclined, or superinclined, depending on the freestream Mach number and the angle the surface makes with the freestream vector. Because of their inclination, superinclined surfaces fall entirely outside of their own domain of influence. This means that boundary conditions cannot be imposed on the upstream side of superinclined surfaces, and these surfaces must simply ingest whatever flow is incident on them.

As such, superinclined panels are used to cancel out incident flow on permeable surfaces, such as engine inlets and outlets – surfaces which in reality ingest and expel flow but whose inner geometries are too complex to model directly [90,121]. Doing so allows for the required inner flow properties to be achieved aft of permeable surfaces [2, 50, 90, 121].

2.2.1 Difficulties with Superinclined Panels

A review of the literature regarding the implementation and use of supersonic panel methods reveals that difficulties surrounding the use of superinclined panels abound. Kroo et al. report having to model flow-through nacelles as parabolic bodies of revolution when using PAN AIR, instead of being able to close them with superinclined panels [14]. Chan reports having the same issue and states that closing an engine nacelle with superinclined panels with specified normal fluxes only worked in subsonic flow [18]. Giblette successfully modeled a configuration with superinclined panels [19]; however, these panels were at the most-downstream part of the configuration and did not influence any other part of the flow. Rajarayan et al. report that the presence of superinclined panels on their configuration being automatically optimized caused PAN AIR to fail [93]. The developers of

ZONAIR recommend that superinclined panels be avoided and present some methods for circumventing this [82].

However, anyone involved with the development of PAN AIR or employed by Boeing reports no issues with superinclined panels. Madson and Erickson report that the recommended boundary conditions in the PAN AIR user’s manual [124] for superinclined panels resulted in a singular system matrix; instead, enforcing zero perturbation potential and zero normal reduced current velocity produced a valid result [90]. Tinoco et al. [91], Ehlers et al. [2], Cenko et al. [89], and Herling et al. [92] all report successfully using superinclined panels to close engine inlets and outlets in supersonic flow.

Hence, superinclined panels are useful but require a more thorough treatment. In this work, the calculation of potential influence coefficients for higher-order superinclined panels is presented. However, more work needs to be done in order to fully understand and be able to implement superinclined panels in the pilot code. This includes determining correct boundary conditions for superinclined panels and determining the correct method for calculating velocity influence coefficients for both sub- and superinclined panels. In addition, the superinclined influence coefficient procedure presented here needs to be verified as part of a functional panel method.

2.2.2 Boundary Conditions on Superinclined Panels

As stated previously, boundary conditions cannot be imposed on the upstream surfaces of superinclined panels [21,33]. To do so would require changing the singularity distributions on surfaces upstream of the superinclined panels, which violates the principle that disturbances only propagate downstream in linearized supersonic theory. Rather, superinclined panels must simply absorb any flow incident on their upstream surface.

In a general sense, boundary conditions on a superinclined surface constitute an initial condition for the two-dimensional wave equation (a second-order PDE) [21,41]. Because of this, two boundary conditions are required in order to fully define the flow downstream of the superinclined surface (i.e. both velocity potential and its derivative must be specified on the surface) [21,41].

In the case of the Morino formulation, it has been reported that the correct boundary conditions to impose on the downstream surface of a superinclined panel are zero perturbation potential and zero normal reduced current velocity (i.e. the conormal derivative of velocity potential, $\frac{\partial \phi}{\partial n}$) [90]. Together, these ensure the inner perturbation potential remains zero, as required for the Morino formulation. These boundary conditions may be applied to superinclined panels that point both upstream (i.e. the flow downstream of them is inside the configuration) and downstream (i.e. the flow downstream of them is outside the configuration). With this, the flow expelled by superinclined panels pointing into the flow would simply be equal to the freestream flow.

For the source-free formulation, the downstream conditions would also be dictated by the required inner potential for panels that point upstream. For panels that point downstream, the condition of zero perturbation potential and zero normal reduced current velocity would likely again need to be used, as with the Morino formulation.

When considering the source-free formulation, an issue with superinclined panels appears. In the source-free formulation, the conormal derivative of the perturbation flow is set by specifying the inner potential. Since there are no sources, there is no discontinuity in conormal derivative across the panels, and the zero-mass-flux boundary condition is satisfied. However, this cannot be the case (in general) with superinclined panels. Because superinclined panels exert no influence on their upstream surface, the conormal derivative on the upstream surface is fixed by the upstream flow. If no source distribution is present, this derivative is then imposed on the downstream surface. Thus the desired inner potential cannot be enforced below superinclined panels without sources. It may be possible to distribute sources only on the superinclined panels. This would allow for a jump in the conormal derivative such that the desired inner potential could still be achieved while maintaining some of the speed advantage of the source-free formulation.

Once the correct boundary conditions are determined, the source and doublet strengths must be solved for. The flow incident on superinclined panels may be dependent upon upstream singularity distributions (e.g. inlets to wing-mounted engines that are within the

domain of influence of the forward fuselage). Since the superinclined panels must cancel out this incident flow, the distributions of both doublets and sources on these panels are not known a priori. Thus, these source strengths must be solved for in the linear system of equations.

For both the Morino and source-free formulations, extra control points would need to be placed in order to account for the unknown source strengths in the linear system of equations. At these extra control points, the extra boundary conditions would be enforced. The fact that sources would need to be placed on superinclined panels in the source-free formulation would negate some of the computational efficiency of the source-free formulation, since source influences would then need to be calculated at all control points. However, this would only be necessary for the superinclined panels, making the source-free formulation still less expensive than the Morino formulation.

As described by Ehlers et al. [2], if panels are used to close the outlets of engines and these panels have a specified mass outflux, then they must also be surrounded by a shed wake surface. This is because the flow coming out of these panels will have a different total head than the rest of the flow. This difference in head is the reason for the existence of the shear layer of an exhaust plume. A wake sheet is the potential-flow approximation of this shear layer [21]. Thus, the flow coming out of the engine must be separated from the rest of the flow.

2.2.3 Calculation of Velocity Influence Coefficients

As stated above, the boundary conditions for superinclined panels include some condition on the derivative of the velocity potential (i.e. the velocity). Thus, in order to form a linear system of equations, it must be possible to calculate velocity influence coefficients for all panels influencing such control points (not just the superinclined panels). Chapter 5 in this work gives only the calculation of potential influence coefficients. In the literature, the presentation of methods for calculating velocity influence coefficients was found to be even less clear than that for potential influence coefficients. Thus, it is anticipated that de-

termining the correct method for calculating velocity influence coefficients would comprise the bulk of the work necessary to implement superinclined panels.

2.3 Alternative Formulations

The Morino and source-free boundary conditions presented above are not the only ways to specify boundary conditions for a panel method.

2.3.1 Neumann Formulations

The simplest alternatives to the formulations presented above are Neumann formulations. Neumann boundary conditions are expressed in terms of the fluid velocity rather than the velocity potential. For a Neumann formulation, zero flow normal to the surface is specified (either linearized mass flux or true velocity). Enforcing this at a set of control points results in a system of linear equations, as with the Dirichlet formulations presented above. The solution to this system gives the unknown source and doublet strengths.

Neumann formulations may be implemented with just sources (e.g. see [53,87]), just doublets (as with the source-free formulation presented here), or both (PAN AIR allows for such Neumann boundary conditions [33,121]). However, there are trade-offs. For a source-only Neumann formulation, the problem is well-posed, but lifting flows cannot be modeled without the addition of doublet wake panels [53,87]. For a doublet-only Neumann formulation, the problem is ill-posed, as any arbitrary constant may be added to the internal potential (and so to the doublet strengths), and the doublet strengths will still satisfy the prescribed boundary condition. This is often overcome by placing a single panel inside the configuration on which the velocity potential must be zero [50,121]. Doing this makes the problem well-posed, but adds extra complexity. The same goes for a source-and-doublet Neumann formulation, but it has been reported that source-and-doublet formulations give better results than doublet-only formulations [50].

There are some advantages to Neumann boundary condition formulations compared to Dirichlet formulations. One is that, since the inner potential is not taken into consideration,

thin configurations may be modeled. In addition, it has been reported that velocity boundary conditions produce more accurate results for blunt bodies in compressible flow than mass flux boundary conditions [89,121]. Despite these advantages and due to difficulties in deriving velocity influence coefficient calculation methods, the scope of this work is limited to Dirichlet formulations.

A Neumann formulation similar to the Morino formulation was presented by Hunt et al. [40,78,111–114]. Hunt’s formulation is called source and vortex evaluation by relaxation (SAVER). Within SAVER, surface vorticity distributions, rather than doublet distributions, are used along with source distributions. Using source and vortex distributions along with a Neumann boundary condition formulation leads to a well-posed problem.

Like the Morino formulation, SAVER is based on the concept of zero perturbation flow inside the configuration (i.e. $\phi_i \equiv 0$). From this, the source strengths are calculated explicitly as a function of the configuration geometry and the freestream condition. Then, it is recognized that the tangential component of perturbation velocity on the outer surface of the configuration must be equal to the local vorticity. Using an iterative procedure, the surface vorticity is updated until the local vorticity and the induced tangential velocity on the surface match. The efficacy of this method is not known, as Hunt et al. never produced a working code or example results.

2.3.2 Source-Free Formulation for Zero Normal Velocity

Rather than determining a source-free inner potential for zero linearized mass flux, it may also be possible to determine a source-free inner potential that results in zero normal velocity. If sources are not present, then there is no jump in the conormal derivative of velocity potential going from the inside of the configuration to the outside. Thus, the only jump in velocity may be in the tangential direction and will be equal to the local gradient of doublet strength. This allows for enforcement of zero normal velocity using a Dirichlet formulation

Zero normal velocity on the surface of the configuration may be expressed as

$$\nabla \Phi \cdot \mathbf{n} = \nabla (\phi_\infty + \phi) \cdot \mathbf{n} = 0 \quad (2.23)$$

Per the previous discussion, if sources are not present, then

$$\nabla \phi \cdot \mathbf{n} = \nabla \phi_i \cdot \mathbf{n} \quad (2.24)$$

This comes not from the definition of zero source strength, but the fact that doublets cannot induce a jump in the normal derivative of velocity potential. Combining this with Eq. (2.23) results in

$$\nabla (\phi_\infty + \phi_i) \cdot \mathbf{n} = 0 \quad (2.25)$$

This will be satisfied if (sufficient but not necessary)

$$\phi_\infty + \phi_i = C \quad (2.26)$$

where C is an arbitrary constant. For simplicity, let $C = 0$, which results in

$$\phi_i = -\phi_\infty = -\mathbf{r}^T \hat{\mathbf{c}} \quad (2.27)$$

If the inner potential satisfies this condition without sources, then the velocity normal to the surface will be zero. As with the previous formulations, the doublet strengths are determined to satisfy Eq. (2.27).

A version of this formulation was implemented in MachLine early in its development. However, bugs were discovered that motivated a switch to the source-free formulation presented at the beginning of this chapter. These bugs may have been due to other issues with the code base, and so it may be valuable to revisit this formulation in the future.

CHAPTER 3

CONFIGURATION MODELING

Thus, to insure precise doublet matching, a program user must be sure, for every abutment containing a matching edge of a doublet design or wake network, that this edge is a refinement of all the other network edges. Similarly, if the abutment contains a supersonic edge, the leading edge of the most downstream pointing network must be a refinement of the others. Finally, in all other cases, some edge must be a refinement of all the others (recall that, if two edges have identical paneling, each is a refinement of the other). - PAN AIR Documentation [33]

Before the aerodynamic influence coefficient matrix (see Eq. (2.22)) can be assembled and the flow properties determined, the configuration mesh needs to be processed. Meshing is an extremely broad topic and is not treated here. The focus of this work begins after the mesh has been generated.

3.1 Unstructured Meshes

An unstructured mesh is one where the relationships between panels and their arrangement in space are not known a priori. There are two arrangements typically used for storing unstructured meshes. With the first arrangement, the corner locations of each panel in space are given in sequence. In this arrangement, the locations of the vertices of the first panel are listed, then those of the second panel, etc. This arrangement is used in the popular STL file format. The second arrangement for specifying unstructured meshes involves first listing the location of each unique vertex in the mesh without reference to the panels. Then, the corners of each panel are specified by listing the indices of the vertices listed in the previous step that define each panel. For example, the first panel is listed as having its corners at vertices 1, 2, and 3, then the second panel is listed as having its corners at vertices 1, 3, and 6, etc. This arrangement is used in the VTK and TRI file formats.

Within the pilot code, the vertices and panels are stored according to the second format arrangement. An object-oriented approach has been taken, and so a surface mesh object contains arrays of vertex and panel objects with the panels referencing the vertices. This structure lends itself well to parsing meshes from VTK and TRI files. However, the pilot code is also able to read in STL files. This requires the extra step of searching for and collapsing references to duplicate vertices.

3.2 Coordinate Systems

Multiple coordinate systems are used in a compressible panel method. These coordinate systems are used to simplify certain calculations and inputs. Typically, a given quantity is calculated in the most convenient coordinate system and then transformed back to the needed coordinate system.

The first of these coordinate systems is the global or reference coordinate system (denoted by the superscript g). This is the coordinate system of the body mesh file. The second is the compressibility coordinate system (denoted by the superscript c). The compressibility system is a rotation of the global system such that the x^c -axis is aligned with the freestream direction, $\hat{\mathbf{c}}$. The third is the scaled coordinate system (denoted by the superscript s). This is a scaling of the compressible coordinate system in the x^c direction by β or B (depending on the flow condition). Since only a scaling is applied and not a rotation, the axes of the compressibility and scaled coordinate systems remain aligned. It is in the scaled coordinate system that the Prandtl-Glauert equation becomes either Laplace's equation or the wave equation. The fourth is the local coordinate system (denoted by the superscript l). The local coordinate system is a rotation of the global coordinate system into the plane of the panel under consideration. The fifth is the local-scaled coordinate system (denoted by the superscript ls). The local-scaled coordinate system is a rotation of the scaled coordinate system into the plane of the panel under consideration. In both the local and local-scaled coordinate systems, the x -axis is aligned with the projection of the freestream velocity into the plane of the panel. This is especially important for the local-scaled coordinate system. For an incompressible flow, the local and local-scaled coordinate systems are identical.

The local-scaled coordinate system is used in calculating panel influence coefficients. This is because the factor $(1 - M_\infty^2)$ does not appear in the influence calculations when the local-scaled system is used, which simplifies the influence calculations. In addition, the plane of the panel is aligned with the x^{ls} - y^{ls} plane, simplifying surface integrals. For compressible flows, the transformation from global to local-scaled coordinates is not orthogonal, due to the scaling [33].

In the pilot code, only the global and local-scaled coordinate systems are used explicitly. However, it is useful to know the other coordinate systems in order to understand some theoretical aspects of panel methods. In the following sections, the transformations between various coordinate systems will be described.

3.2.1 Transformation from Global to Compressibility Coordinates

The first transformation considered here is that between the global and compressibility coordinate systems. An important parameter defining this transformation is the compressibility axis. This is the scaled axis in the Prandtl-Glauert equation and is typically assumed to be the x axis [33]. However, it is convenient to allow for an arbitrary compressibility direction. If the compressibility axis is aligned with the uniform onset flow, then the compressibility direction is

$$\hat{\mathbf{x}}_c = \hat{\mathbf{c}} \quad (3.1)$$

Within PAN AIR, the transformation from global to compressibility coordinates is defined by a set of angle of attack and sideslip rotations [33]. This method assumes that the angle of attack defines a rotation about the global y -axis. However, it may be that the user has input a mesh where the angle of attack is a rotation about some other axis. As such, this transformation is here extended to arbitrary global coordinate systems.

Let the positive spanwise axis (i.e. pointing right out the side of the configuration as viewed by a forward-facing pilot) expressed in the global coordinate system be denoted

by $\hat{\mathbf{s}}^g$. For example, if the user is using the standard aircraft-fixed coordinate system, the positive spanwise axis is the positive y -axis, and

$$\hat{\mathbf{s}}^g = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (3.2)$$

In the classical literature regarding linear aerodynamics (e.g. see [21]), the z -axis is typically aligned with the freestream. In this case, the positive global x -axis is aligned with the positive spanwise axis.

The transformation to compressibility coordinates consists of a rotation about the spanwise axis and a subsequent rotation about the resulting z -axis [33]. This transformation is shown in Fig. 3.1. In the example shown here, the standard aerodynamic coordinate system is used where the spanwise axis is the positive y -axis.

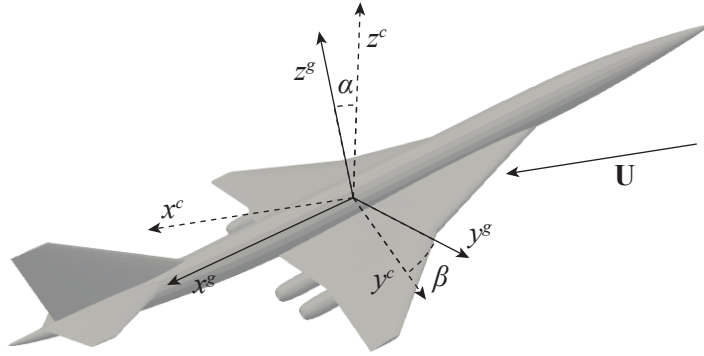


Fig. 3.1: Rotation from the global to the compressible coordinate system.

Rather than relying on aerodynamic angles and sines and cosines thereof, a rotation matrix may be calculated by expressing the compressible coordinate axes in the global coordinate system. The compressible z -axis is perpendicular to both the compressibility axis and the spanwise axis, and so

$$\hat{\mathbf{z}}_c^g = \frac{\hat{\mathbf{c}}^g \times \hat{\mathbf{s}}^g}{\|\hat{\mathbf{c}}^g \times \hat{\mathbf{s}}^g\|} \quad (3.3)$$

where $\hat{\mathbf{z}}_c^g$ is the z -axis of the compressibility frame expressed in the global frame. Completing the right-handed coordinate system gives

$$\hat{\mathbf{y}}_c^g = \hat{\mathbf{z}}_c^g \times \hat{\mathbf{c}}^g \quad (3.4)$$

Thus, the transformation matrix is given by

$$[A_{g \rightarrow c}] = [\hat{\mathbf{c}}^g \quad \hat{\mathbf{y}}_c^g \quad \hat{\mathbf{z}}_c^g] \quad (3.5)$$

3.2.2 Transformation from Compressibility to Scaled Coordinates

The transformation from compressible to scaled coordinates is given by [33]

$$[A_{c \rightarrow s}] = \begin{bmatrix} \frac{1}{1-M_\infty^2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Applying this transformation, the Prandtl-Glauert equation becomes the Laplace equation in subsonic flow or the classical wave equation with a wave speed of 1 in supersonic flow when expressed in the compressibility frame.

3.2.3 Transformation from Global to Local-Scaled Coordinates

The transformation from global to local scaled coordinates is given by [33]

$$[A_{g \rightarrow ls}] = \begin{bmatrix} |\{\mathbf{n}^g, \mathbf{n}^g\}|^{-1/2} ([C]^g \hat{\mathbf{u}}^g)^T \\ \frac{rs}{B} ([C]^g \hat{\mathbf{v}}^g)^T \\ B |\{\mathbf{n}^g, \mathbf{n}^g\}|^{-1/2} (\mathbf{n}^g)^T \end{bmatrix} \quad (3.7)$$

where the inner product $\{\cdot, \cdot\}$ is defined by

$$\{\mathbf{a}, \mathbf{b}\} = \mathbf{a}^T [B] \mathbf{b} \quad (3.8)$$

with the matrix $[B]$ given by Eq. (1.71),

$$[C] = [I](1 - M_\infty^2) + M_\infty^2 \hat{\mathbf{c}} \hat{\mathbf{c}}^T \quad (3.9)$$

and \mathbf{n} is the panel normal vector, r is the panel inclination indicator (Eq. (3.11)), and s is the flow type indicator (1 for subsonic flow, -1 for supersonic flow). The vectors $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ are chosen to appropriately define the coordinate system. Following Epton and Magnus [33], $\hat{\mathbf{v}}$ is a unit vector orthogonal to both the panel normal and the freestream velocity. The vector $\hat{\mathbf{u}}$ is then chosen to complete the right-handed coordinate system with $\hat{\mathbf{v}}$ and \mathbf{n} . Under this definition, $\hat{\mathbf{u}}$ is the unit-length projection of the freestream velocity into the plane of the panel. In the special case that \mathbf{n} is parallel to $\hat{\mathbf{c}}$ (i.e. the panel is perpendicular to the freestream flow), the choice of $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ is arbitrary so long as both are orthogonal to \mathbf{n} .

While the origins of the global, compressibility, and scaled coordinate systems all coincide, the origin of the local-scaled system is the centroid of the panel for which the system is defined. Thus, transforming a point location to the local-scaled system from the global system is done using the formula

$$\mathbf{P}^{ls} = [A_{g \rightarrow ls}](\mathbf{P}^g - \mathbf{P}_{\text{cent}}^g) \quad (3.10)$$

where \mathbf{P}_{cent} is the panel centroid. For direction vectors, the transformation simply involves premultiplying by $[A_{g \rightarrow ls}]$.

In the case of incompressible flow ($M_0 = 0$), $A_{g \rightarrow ls}$ is equivalent to $A_{g \rightarrow l}$, since no scaling occurs.

3.3 Panel Geometric Parameters

When each panel is initialized, several geometric parameters are calculated to save time in subsequent calculations. The most basic parameters, such as the panel area and normal vector, \mathbf{n} , are calculated using standard formulas. The outward normal vector for each panel edge is also calculated, both in global and local-scaled coordinates. The transformation matrix, $A_{g \rightarrow ls}$ is calculated once for each panel.

The panel inclination indicator, r , is calculated from the panel normal vector, \mathbf{n} , using the relation [33]

$$r = \text{sign}(\{\mathbf{n}^g, \mathbf{n}^g\}) \quad (3.11)$$

where the inner product $\{, \}$ is given by Eq. (3.8). When $r = 1$, the panel is subinclined, and when $r = -1$, the panel is superinclined (see Section 1.2.3). A panel is Mach-inclined if

$$\{\mathbf{n}^g, \mathbf{n}^g\} = 0 \quad (3.12)$$

Mach-inclined panels are not allowed. If the panel is Mach-inclined, then the transformation from global to local-scaled coordinates is undefined (see Eq. (3.7)).

3.4 Edge Search and Characterization

Once all panels have been initialized, a search is made to locate adjacent panels. This information is then used in enforcing continuity of doublet strength across panel edges and for automatically generating wakes. This is done in the pilot code by looping through every possible pair of panels in turn to check whether they share two vertices. Since panels and vertices are stored separately in the pilot code, with each panel referencing the vertices which define it, adjacent panels can be identified using the vertex indices, as opposed to checking the vertex locations. When a pair of adjacent panels has been located, the information for the edge shared by the two panels is stored in a separate edge data type.

In all types of flow, the edges from which a wake can be shed must be found. Wakes are shed from “sharp” edges. How sharp an edge must be to shed a wake is determined by a user-specified threshold angle. If the flow-turning angle caused by the two panels defining an edge is greater than this threshold, then the edge will shed a wake.

The cosine of the flow-turning angle between two panels is given by

$$\cos \theta = \mathbf{n}_i \cdot \mathbf{n}_j \quad (3.13)$$

for panels i and j . The most the cosine of the flow-turning angle can be is 1, indicating the surface is flat, as shown in Fig. 3.2a. Thus, for some given angle threshold, the cosine of the flow-turning angle must be less than the cosine of the angle threshold to qualify as wake-shedding.

However, this is not sufficient for an edge to be wake-shedding. This is because the sign of the cosine of the flow-turning angle will be the same whether the two panels turn the flow into or away from the mesh. Clearly, only an edge where the orientation of the two panels turns the flow towards the mesh will shed a wake (the opposite condition is shown in Fig. 3.2b). Thus, a second condition can be imposed. If the edge proceeds in a counter-clockwise direction about the normal vector for panel i , then the panels turn the flow into the mesh if the inner product of the edge vector and $\mathbf{n}_i \times \mathbf{n}_j$ is positive.

There is one final condition that must be imposed. As shown in Fig. 3.2c, both previous conditions may be met on an edge where the wake would be shed into the body, as on a sharp leading edge. Hence, at least one of the panel normals must also point downstream. If all three conditions are met, then that edge is marked as wake-shedding.

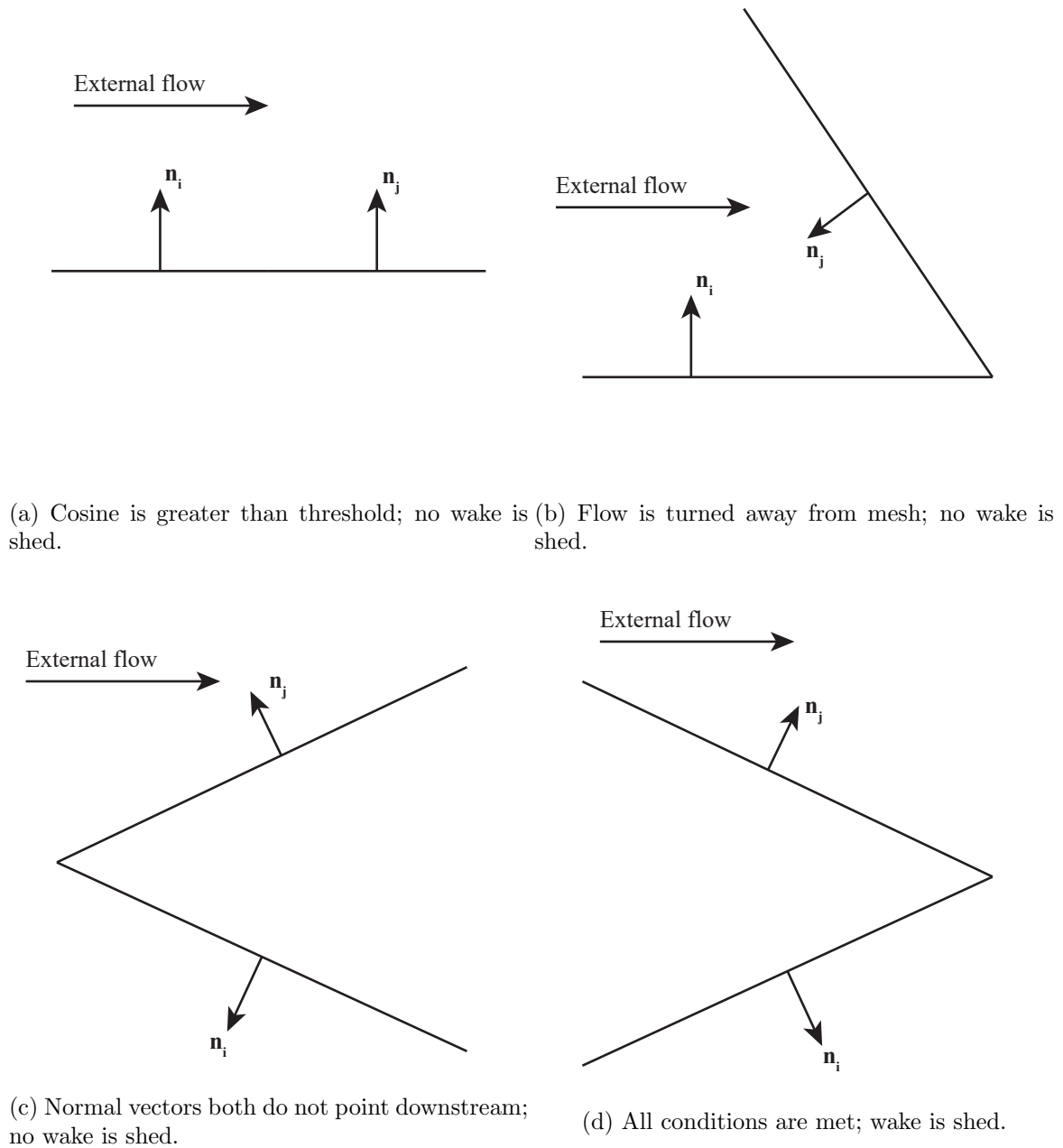


Fig. 3.2: Side views of different edge cases showing when a wake will and will not be shed.

3.5 Wake Modeling

Once the wake-shedding edges have been located, all vertices shared by these edges are identified. Beginning at these vertices and extending in the freestream direction, a set of evenly-spaced wake vertices are created. The number of vertices and how far they

extend in the freestream direction are specified by the user. For steady flows without wake relaxation, the wake can be accurately modeled using only one discretization in the streamwise direction, as the wake strength is constant in the streamwise direction. Then, using the vertices just created, a set of panels is created to fill in the wake. An example of this is shown in Fig. 3.3. Each set of four wake vertices defines two triangular wake panels, since the pilot code can only handle triangular panels. The total distance that the wake extends aft of the aircraft origin is referred to here as the Trefftz distance and is set by the user.

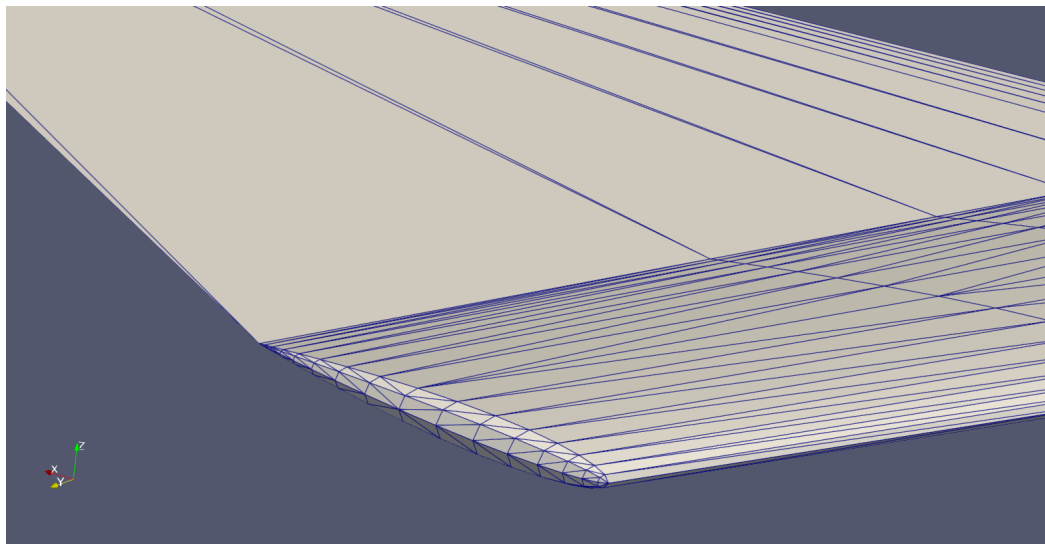


Fig. 3.3: Wake panels shed from the trailing edge of a swept wing, illustrating the basic wake-generation algorithm in MachLine.

Once wake panels have been generated, it is necessary to “un-stitch” the mesh along wake-shedding edges. This is because the unknown doublet strengths are defined at the mesh vertices, and there is a jump in doublet strength across the wake. Thus, an extra vertex needs to be added at each vertex along a chain of wake-shedding edges. The original vertex will store the doublet strength for the top surface and the new vertex will store the doublet strength for the bottom. This is done in MachLine by adding clones of the vertices along the wake-shedding edges to the mesh and then updating the panel objects to reference the new cloned vertices as appropriate. Within MachLine, panels on top of the wake-shedding

edge continue referencing the original vertices while the bottom panels reference the new cloned vertices. For wake panels, since their doublet strength is the difference of the doublet strengths on the top and bottom of the wake-shedding edge, references to both original and cloned vertices are maintained.

As currently implemented, the wake model in the pilot code cannot properly model wake-body interactions. The reason for this is discussed in Section 2.1. More work is necessary to be able to capture these interactions (see Section 10.6).

3.5.1 Automatic Supersonic Wake Truncation

In supersonic flows, much of the wake does not need to be explicitly modeled as it will have no influence on upstream points. In legacy panel codes, the user is required to manually specify the wake [33], and so accounting for this lack of interaction is up to the user. However, it is possible to truncate supersonic wakes automatically.

The first step of this automatic truncation algorithm is determining the most downstream vertex on the mesh. The most downstream point is the one whose x coordinate expressed in the compressible coordinate frame is the greatest. This coordinate, x^c , is equal to

$$x^c = \mathbf{r}^g \cdot \hat{\mathbf{c}}^g \quad (3.14)$$

where \mathbf{r}^g is the vertex location expressed in the global frame. Since no part of the body mesh will be influenced by any points further downstream than the point just identified, the wake may be truncated at that distance. Thus, x^c just determined replaces the Trefftz distance for specifying the wake.

3.6 Mesh Mirroring

In incompressible flows, the influence of a panel on a given point will be the same as the influence of that panel mirrored across some plane on the point mirrored across the same plane. For compressible flows, the influence coefficients are the same for both cases if the freestream vector also lies in the plane across which the geometry is being mirrored. In addition, if a mesh is symmetric about some plane and the freestream vector lies in that plane, then the resulting flowfield will also be symmetric across that plane. Hence, any symmetry in the mesh may, depending on the flow condition, be exploited to speed up a panel method.

There are three possible symmetry cases, shown in Fig. 3.4, that are implemented in MachLine. For the first case, the user gives a full mesh (symmetric or not), and no mirroring is done. In the other two cases, the user gives a half mesh and specifies that it is to be mirrored across a given plane (e.g. the user has a mesh for the right half of a typical airplane and specifies the mesh is to be mirrored across the $x - z$ plane). What distinguishes these two cases is the flow condition. A *symmetric flow*, with respect to the mesh, is one where the freestream velocity vector falls in the plane of symmetry. A flow is *asymmetric*, with respect to the mesh, otherwise. Thus, the second symmetry case is where the mesh is mirrored and the flow is symmetric. The third symmetry case occurs when the mesh is mirrored and the flow is asymmetric.

In the second case (i.e. mirrored mesh and symmetric flow condition), the computation only needs to occur on half of the mesh as the results (source and doublet strengths, surface pressures, etc.) will be symmetric. This effectively halves the size of the problem to be solved, leading to significant computational savings. However, the mirrored half of the mesh still exerts an influence on the existing half. To calculate this influence, the influence of each existing panel on a mirror of the control point under consideration is calculated. This gives the same result as calculating the influence of a mirrored panel on the existing control point but is much simpler and less memory-intensive to implement. This influence is added to the influence of the original panel on the original control point in the AIC matrix.

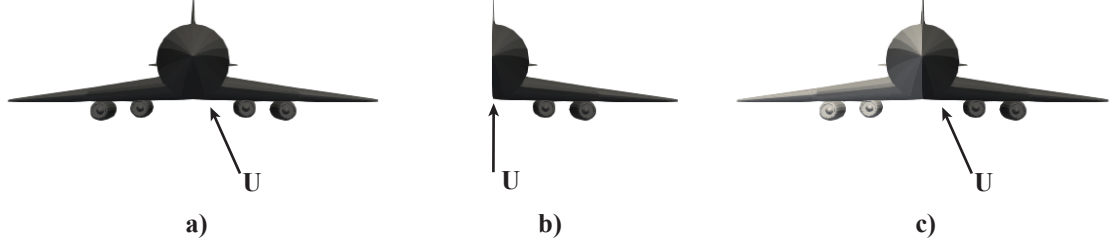


Fig. 3.4: The three possible cases for mesh mirroring in MachLine: a) Case 1: no mirroring, b) Case 2: mirrored mesh and symmetric flow condition where the mirrored half of the mesh does not need to be explicitly modeled, and c) Case 3: mirrored mesh and asymmetric flow condition where the mirrored half of the mesh is explicitly modeled and has different flow results from the original half.

In the third case (i.e. mirrored mesh and asymmetric flow condition), the computation needs to occur on both the original and mirrored halves of the mesh, since the results will be asymmetric. For incompressible flows (i.e. $M_\infty = 0$), the influence of a panel on a control point is independent of the freestream direction. In this case, the savings from calculating mirrored influence coefficients using an existing panel and a mirrored control point may be maintained. However, for compressible flows, the panels must be mirrored explicitly and the influence of these mirrored panels calculated accordingly. Within MachLine, extra geometric parameters are calculated during initialization which define the mirror of each panel. When the function is called to calculate the influence of a panel on a given point, one of the arguments is whether the original or mirrored geometry is to be used.

3.7 Control Point Placement

As discussed in Section 2.1, certain conditions must be enforced on the inner velocity potential in order to determine the distributions of source and doublet strength on the configuration surface. These conditions are enforced at a set of control points placed inside the configuration. For the formulations considered in this work, there need to be as many control points inside the configuration as there are unknown doublet strengths. For both

linear and quadratic doublet distributions (see Chapter 5), this means one control point for each vertex in the mesh. From a purely theoretical standpoint, these control points may be placed anywhere inside the configuration [67]; however, practice has shown that it is best to have them very close to the configuration surface [7].

It is possible to place control points directly on the configuration surface for an inner Dirichlet formulation. Such was done in PAN AIR [2,33]. However, calculating the induced potential at such points then involves knowing the solid angle of the surface at such points, since the sphere used to exclude point \mathbf{P} from the flow region in deriving Green's third identity is not full [43]. For control points placed within a panel (as was done in PAN AIR), this is relatively simple, as the solid angle is just 2π since only a half sphere is needed. However, for control points placed at vertices, this requires some significant computation to determine the solid angle subtended by the mesh at that vertex. It is simpler to just place the control points fully within the surface and assess the sensitivity of the solutions to the offset distance.

In MachLine, a control point is placed just inside the configuration from each vertex according to

$$\mathbf{P}_c = \mathbf{P} - k_1 \mathbf{d} \quad (3.15)$$

where \mathbf{P}_c is the control point location, \mathbf{P} is the associated vertex location, k_1 is a non-dimensional control point offset, and \mathbf{d} is a unit vector giving the offset direction for the control point. Typically, \mathbf{d} should point outward from the surface and be normal to the surface in an average sense (average because neighboring panels are not necessarily coplanar).

Originally, control points were placed in MachLine using the formula [7,67,117,125]

$$\mathbf{P}_c = \mathbf{P} - k_1 l_{\text{avg}} \mathbf{d} \quad (3.16)$$

where l_{avg} is the average length of the edges adjacent to the vertex. This is following the method proposed by Davis [7, 68], which was taken from Maruyama et al. [67]. However, the reason for scaling the control point locations based on some mesh-dependent length was never given by either Davis or Maruyama. Through various numerical studies, it was found that Eq. (3.15) is more appropriate as this leads to the recommended range for k_1 being independent of the mesh and flow condition. These studies are presented in Section 8.1.

Originally, the unweighted average of the adjacent panel normals was used for \mathbf{d} . However, this unweighted average led to poor results. In particular, the doublet strength and surface pressure at the trailing edge of wings with uneven paneling showed unrealistic fluctuations. To address these trailing edge fluctuations, the calculation of the average normal vector was weighted by the angle of each panel corner at that vertex. This angle is given by

$$\theta_{\text{corner}_i} = \arccos(-\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_{i-1}) \quad (3.17)$$

where $\hat{\mathbf{n}}_i$ is the in-plane outward normal vector for edge i . There is no need to calculate the sum of these corner angles because \mathbf{d} is always normalized.

3.7.1 Control Points on Wake-Shedding Edges

Along wake-shedding edges, control points are placed relative to each vertex and cloned vertex. If the above method were to be used as is for these control points, then the control points placed relative to cloned vertices would be the same as the control points placed relative to the original vertices, leading to a singular AIC matrix since the rows corresponding to cloned control points would be identical. Hence, a more involved method is needed for placing control points along wake-shedding edges. These control points need to be unique, inside the mesh, and placed in a regular fashion so as to avoid solution fluctuations. To meet these requirements, the following method was developed.

For a given vertex attached to at least two wake-shedding edges, its neighboring panels may be divided into as many contiguous groups, as shown in Fig. 3.5. The groups are

separated by the wake-shedding edges. One control point needs to be placed for each group. For each group of contiguous panels, the average tangent vector of the two bounding wake-shedding edges, denoted by $\hat{\mathbf{t}}_a$, is first determined. Then, a vector perpendicular to $\hat{\mathbf{t}}_a$ that also lies inside one of these panels needs to be found for each group of panels. This vector is denoted by $\hat{\mathbf{t}}_p$, and it is assumed that $\hat{\mathbf{t}}_p$ is unique. The vector $\hat{\mathbf{t}}_p$ may be found by looping through each panel in the group, projecting an arbitrary vector from the panel's surface onto the orthogonal complement of $\hat{\mathbf{t}}_a$ [126], and checking whether this projection still lies inside the panel from which it originated. If this arbitrary vector is denoted by \mathbf{v} , then its projection onto the orthogonal complement of $\hat{\mathbf{t}}_a$ is given by

$$\mathbf{v}_p = ([I] - \hat{\mathbf{t}}_a \hat{\mathbf{t}}_a^T) \mathbf{v} \quad (3.18)$$

where $[I]$ is the identity matrix. If the projection does lie inside its parent panel, it is assigned to $\hat{\mathbf{t}}_p$.

The vector $\hat{\mathbf{t}}_p$ now defines a direction perpendicular to the average of the wake-shedding edges that also lies on the surface of the neighboring panels. It thus represents an outer bound for where the control point may be placed. Combining this with the angle-weighted normal vector, \mathbf{n}_{avg} , for all the panels in this group, the control point may be robustly placed within the mesh. This is done by placing the control point using $\hat{\mathbf{t}}_p$ and then offsetting it from the surface opposite the direction of \mathbf{n}_{avg} , as in

$$\mathbf{d} = \hat{\mathbf{t}}_p - R_{\text{offset}} \mathbf{n}_{\text{avg}} \quad (3.19)$$

Here, the parameter R_{offset} is chosen such that the control point will be placed approximately halfway between the outer surface of the mesh and the midplane between the two closest panels touching the vertex. This is shown in Fig. 3.6. Each pair of panels neighboring the vertex is considered to find the minimum cosine of the angle between two panel normals, $C_{\theta_{\min}}$. The angle ϕ is the angle subtended by R_{offset} . For the control point

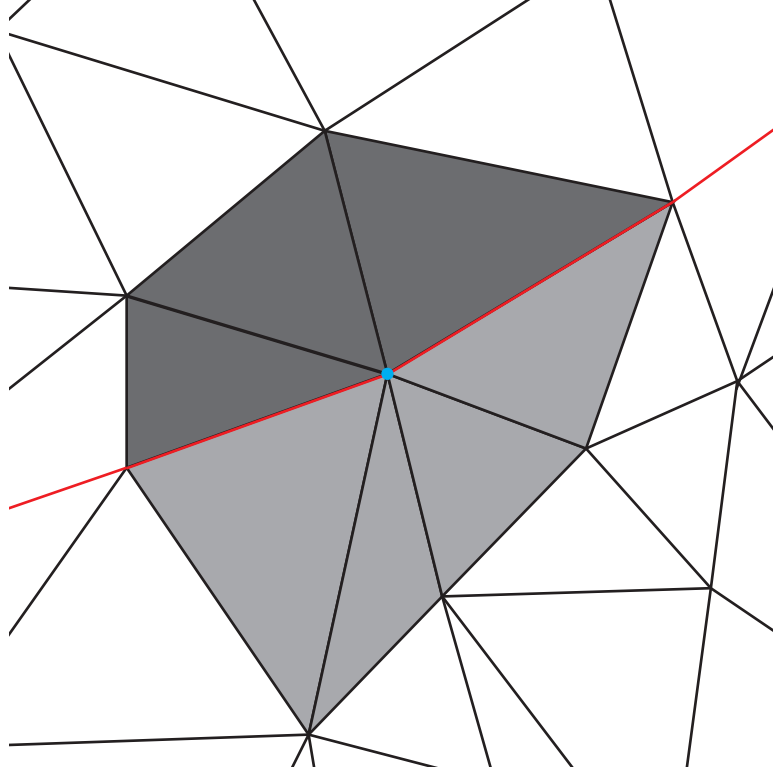


Fig. 3.5: The two contiguous groups of panels (light gray and dark gray) touching the vertex highlighted in blue on either side of a string of wake-shedding edges (red).

to be halfway between the surface and the midplane, it should be that $\phi = \theta/4$. It may then be deduced that

$$2R_{\text{offset}} = \sin(2\phi) = \sin\left(\frac{\theta}{2}\right) \quad (3.20)$$

Using a half-angle identity and knowing $\cos \theta = -\cos \theta_{\min}$, this will become

$$R_{\text{offset}} = \frac{1}{2} \sin\left(\frac{\theta}{2}\right) \quad (3.21)$$

$$R_{\text{offset}} = \frac{1}{2} \sqrt{\frac{1 + \cos \theta_{\min}}{2}} \quad (3.22)$$

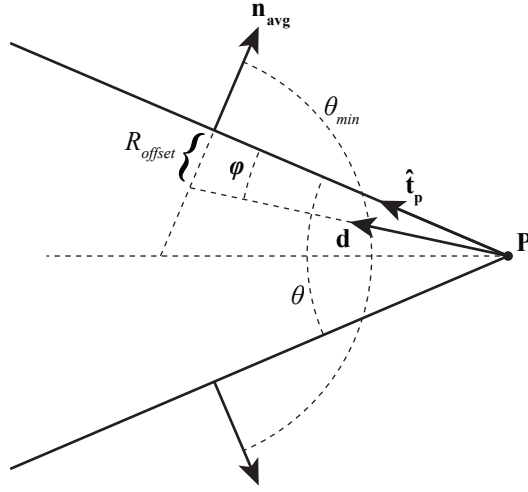


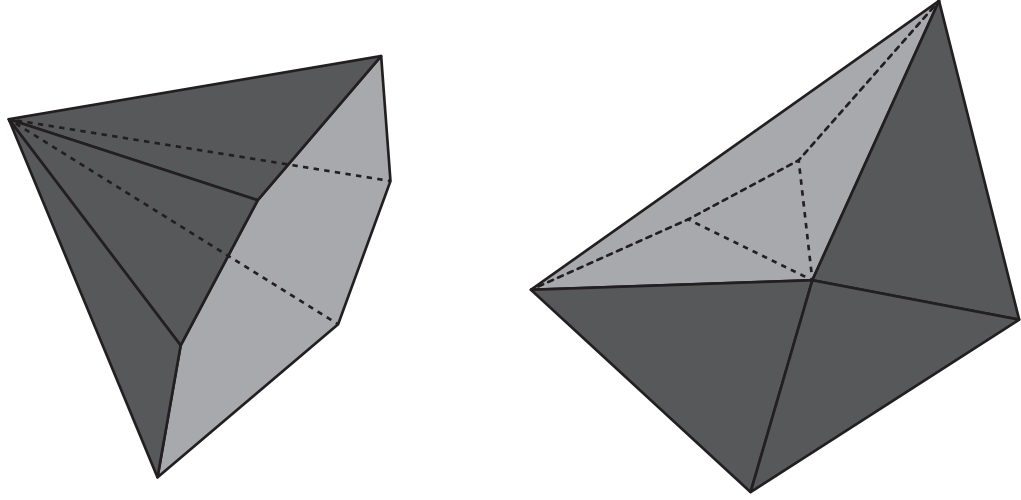
Fig. 3.6: Placement of control points for vertices belonging to wake-shedding edges.

3.7.2 Control Point Placement Correction

Even with these methods, it was discovered that some control points could be placed outside the mesh. This would occur for vertices at which the surrounding panels were non-convex. Here, a convex set of panels is one where none of the panels belongs to the interior of the convex closure of the panels. Examples of convex and non-convex sets of panels are shown in Fig. 3.7. For vertices whose surrounding panels are non-convex, it is possible for the opposite of the average normal vector to not point into the mesh. Hence, methods were needed for automatically detecting control points outside the mesh and correcting their location.

In order to determine if a given control point is outside of the mesh, a simple ray-casting algorithm is used. This algorithm involves drawing a line between a point known to be outside the mesh and the control point of interest. If this line intersects the mesh an even number of times (assuming the mesh is watertight), then the control point must also be outside the mesh. Otherwise, the control point is inside the mesh.

Typically, this algorithm requires checking for intersections between the line and every panel in the mesh, an $O(N)$ operation. To reduce computational expense, the starting point



(a) Convex panels

(b) Non-convex panels

Fig. 3.7: (a) Convex and (b) non-convex sets of panels. For the convex set of panels, the panels (dark gray) are all on the boundary of their convex closure (the boundaries of the convex closure not belonging to panels are shaded light gray). This is not the case with the non-convex set as the panels outlined by the dashed lines are not on the boundary of the convex closure.

(known to be outside the mesh) was chosen as a point offset from the centroid of one of the panels neighboring the vertex by the characteristic length of that panel. With the starting point so close to the control point, only the panels touching the vertex associated with the control point need to be considered. Whether the line between the point outside the mesh and the control point intersects a given panel then needs to be determined.

To do this, let the line connecting the point outside the mesh and the control point under consideration be denoted by

$$\mathbf{r}(s) = \mathbf{P}_{\text{start}} + s\mathbf{d} \quad (3.23)$$

where

$$\mathbf{d} = \mathbf{P}_c - \mathbf{P}_{\text{start}} \quad (3.24)$$

and $\mathbf{P}_{\text{start}}$ is the starting point outside the mesh. Let s^* denote the location in s where the line intersects the plane of the panel. At s^* , it must be that

$$(\mathbf{r}(s^*) - \mathbf{P}_{\text{centroid}}) \cdot \mathbf{n} = 0 \quad (3.25)$$

where $\mathbf{P}_{\text{centroid}}$ is the centroid of the panel and \mathbf{n} is the normal vector to the panel. This may be manipulated into the form

$$(\mathbf{P}_{\text{start}} + s^* \mathbf{d}) \cdot \mathbf{n} = \mathbf{P}_{\text{centroid}} \cdot \mathbf{n}$$

$$s^* = \frac{\mathbf{P}_{\text{centroid}} \cdot \mathbf{n} - \mathbf{P}_{\text{start}} \cdot \mathbf{n}}{\mathbf{d} \cdot \mathbf{n}} \quad (3.26)$$

If the denominator is zero, then the line is parallel to the plane of the panel, and there will not be an intersection.

For the ray-casting algorithm, the point of intersection must lie between the control point and $\mathbf{P}_{\text{start}}$. Otherwise, the intersection of the line with a panel says nothing about whether the control point is outside the mesh. Thus, once s^* is determined, a check is made to see if $0 < s^* < 1$. If this is not the case, then the intersection does not count for the ray-casting algorithm, and computation proceeds to the next neighboring panel.

If the denominator of Eq. (3.26) is not zero and $0 < s^* < 1$, then the location of intersection (denoted by \mathbf{r}^*) may be found by plugging s^* back into Eq. (3.23). It must then be determined whether \mathbf{r}^* actually belongs to the surface of the panel (rather than simply being in the same plane). Let $\hat{\mathbf{n}}_i$ denote the outward, in-plane normal vector for each edge i of the panel. In order for the point of intersection to be on the panel, it must be that

$$(\mathbf{r}^* - \mathbf{P}_i) \cdot \hat{\mathbf{n}}_i \leq 0 \quad \forall i \quad (3.27)$$

where \mathbf{P}_i is the i -th vertex of the panel.

If it has been determined that a control point is outside the mesh, its location must be corrected. This is done by partially reflecting the control point offset vector across a plane of the mesh, and then rescaling this offset vector to maintain the control point offset (k_1). First, each panel touching the vertex is considered to see whether the control point is above the plane of the panel. This is done simply by checking the z coordinate of the control point location expressed in the local frame for each panel, z^l . This is equal to the dot product of the control point displacement from the panel and the panel normal vector. If z^l is positive, then the control point is above that panel.

A sum is then taken of the normal vectors of all the panels the control point is above. Normalizing this vector sum then gives an average normal vector, \mathbf{n}_{avg} , defining the plane across which to reflect the control point offset vector.

Let the control point offset vector be given by

$$\mathbf{d} = \mathbf{P}_c - \mathbf{P} \quad (3.28)$$

where \mathbf{P} is the vertex associated with the control point. The new offset vector, denoted by \mathbf{d}' , is then obtained by partially reflecting \mathbf{d} across the plane defined by \mathbf{n}_{avg} , as in

$$\mathbf{d}' = \mathbf{d} - k_3 \mathbf{n}_{\text{avg}} (\mathbf{d} \cdot \mathbf{n}_{\text{avg}}) \quad (3.29)$$

where k_3 is called the *reflection coefficient*. If $k_3 = 2$, then \mathbf{d}' is a true reflection of \mathbf{d} across the plane defined by \mathbf{n}_{avg} . However, it was found that $k_3 = 2$ often led to the control point simply ending up outside the other side of the mesh. In practice, $k_3 = 1.1$ was found to reliably place control points inside the mesh. The new control point location is then simply

$$\mathbf{P}_c = \mathbf{P} + k_1 \frac{\mathbf{d}'}{\|\mathbf{d}'\|} \quad (3.30)$$

where k_1 is the control point offset parameter defined previously. The division by the magnitude of \mathbf{d}' is necessary to maintain the desired offset between the control point and the vertex.

For robustness, it is best to check the corrected control point location to see if it is once again outside the mesh. If this is the case, then the above method may be reapplied until the control point is inside the mesh. The use of this iterative approach is another motivation for setting $k_3 < 2$. If each iteration applied a full reflection, then the control point could simply bounce back and forth between locations outside the mesh.

CHAPTER 4

DOMAIN OF DEPENDENCE CALCULATIONS

[This] shows that the value of $[\phi]$ does depend on the values of $[\phi]$ and its derivative] inside the cone... This means that Huygens' principle is false in two dimensions. For instance, when you drop a pebble onto a calm pond, surface waves are created that (approximately) satisfy the two-dimensional wave equation with a certain speed c , where x and y are horizontal coordinates. A water bug whose distance from the point of impact is δ experiences a wave first at time $t = \delta/c$ but thereafter continues to feel ripples. These ripples die down...but theoretically continue forever. - Walter A. Strauss [41]

As discussed previously, supersonic flows yield domains of dependence and influence for every point in the flow. For a given control point \mathbf{P}_c , it must be determined whether a given panel falls within \mathbf{P}_c 's domain of dependence (denoted by D_{P_c}). The domain of dependence calculations given in Section J.3 of [33] have been implemented in the pilot code with some modifications.

4.1 Point Domain of Dependence Formulas

For a point \mathbf{Q} to fall within D_{P_c} , it must be upstream of \mathbf{P}_c and within the upstream Mach cone. This first condition can be expressed as

$$(\mathbf{P}_c - \mathbf{Q}) \cdot \hat{\mathbf{c}} \geq 0 \tag{4.1}$$

and the second can be expressed as

$$[\mathbf{P}_c - \mathbf{Q}, \mathbf{P}_c - \mathbf{Q}] \geq 0 \tag{4.2}$$

where the inner product $[\cdot, \cdot]$ is defined by

$$[\mathbf{a}, \mathbf{b}] = \mathbf{a}^T [C] \mathbf{b} \quad (4.3)$$

where the matrix $[C]$ is given by Eq. (3.9).

4.2 Panels and Domains of Dependence

To calculate the influence of a panel on a given control point, it must be known which panel vertices and edges fall within the control point's domain of dependence. For superinclined panels, it must also be known whether the intersection of the panel surface and the domain of dependence is a subset of the panel surface without any edges or vertices falling inside the domain of dependence [2, 7, 33].

Within the pilot code, the domain of dependence is first checked for each mesh vertex, then edges, and then for the entire panel; this method is efficient given the mesh is unstructured. Checking whether a vertex falls within the domain of dependence is straightforward, and this is first done for each of the vertices in the mesh. Each edge is then considered. If at least one edge endpoint is in the domain of dependence, then that edge intersects the domain of dependence. For supersonic edges (i.e. edges inclined to the freestream at angles greater than the Mach angle), the edge may still intersect the domain of dependence even if both endpoints do not. In this case, \mathbf{Q} falls within what is called the *Mach wedge* region of the edge [2, 7]. This region is shown in Fig. 4.1. All points within the Mach wedge are influenced by the edge, but neither of its endpoints.

Finally, even if all vertices and edges of a panel fall outside the domain of dependence, the panel may still belong to the domain of dependence if the panel is superinclined. Such a situation is shown in Fig. 4.2. Equations for detecting Mach wedge conditions and the case where the panel entirely encompasses the domain of dependence are given in [33] and reproduced in MachLine [118].

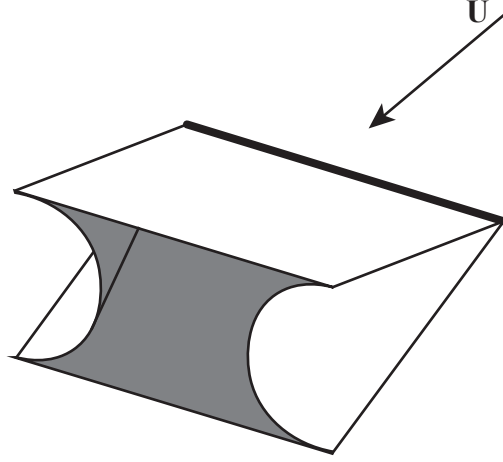


Fig. 4.1: Mach wedge region of a supersonic edge. The Mach wedge is the region in which a point may be influenced by the edge but falls outside the Mach cones for either endpoint.

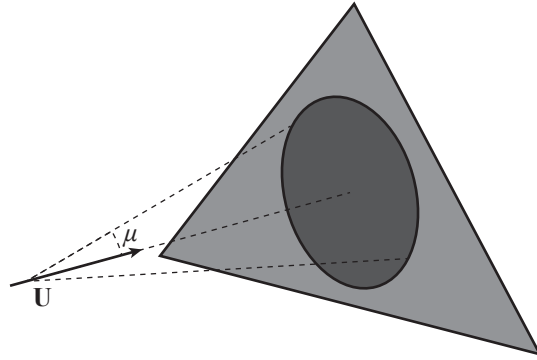


Fig. 4.2: A case where a superinclined panel belongs to the domain of dependence of a control point when none of its vertices or edges do.

4.3 Efficient Domain of Dependence Calculation

The approach described above may be implemented in a simple, brute-force manner. That is, for every control point (or point being influenced), every vertex in the mesh is checked to determine if it falls within the domain of dependence. When determining how N vertices fall relative to the domains of dependence for N control points, this is an $O(N^2)$ operation.

However, it is also possible to calculate point domains of dependence using a simple recursion principle. Within linearized theory, the Mach number is assumed constant throughout the flow. Because of this, all Mach cones have the same half angle. Consider

two points, \mathbf{P} and \mathbf{Q} . If \mathbf{Q} falls inside the domain of dependence of point \mathbf{P} , D_P , then D_Q will be a strict subset of D_P . An example of this is shown in Fig. 4.3.

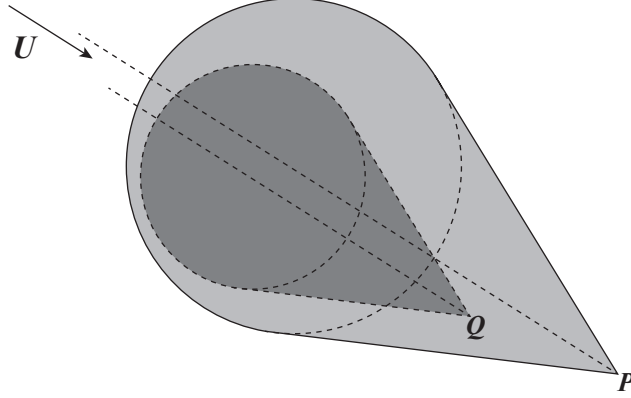


Fig. 4.3: Diagram of the domains of dependence for two points showing how the domain of dependence of one point is entirely encompassed by that of the other point.

Because of this principle, if it is known which vertices belong to D_Q , then it is instantly known that all those points belong to D_P . They do not need to be considered separately. This allows for a more efficient algorithm for determining domains of dependence over the entire mesh. Such an algorithm is developed here.

Assume first that all vertices in the mesh have been sorted so that they proceed in order strictly downstream. This guarantees that the first vertex in the mesh has no vertices in its domain of dependence, as all other vertices are downstream of it, or belong to the same plane perpendicular to the freestream direction. This sort can be accomplished based on the metric

$$x_i = \mathbf{P}_i \cdot \hat{\mathbf{c}} \quad (4.4)$$

where \mathbf{P}_i is the location of vertex i . If for two vertices, i and j , x_i is greater than x_j , then vertex i is downstream of vertex j . Using an efficient sorting algorithm, performing this operation is $O(N \log N)$.

Once this sort has been performed, the domain of dependence for the first vertex is known, because no other vertices can be in its domain of dependence. For all vertices, any downstream vertices cannot be within the domain of dependence.

The algorithm then proceeds to the next vertex upstream. For this second vertex, only the most-downstream vertex (the first one considered) could possibly be within its domain of dependence, and so only this vertex needs to be checked. Whether that vertex falls within the current domain of dependence may be found from Eqs. (4.1) and (4.2). For each successive vertex (proceeding upstream), the vertices downstream of it are checked beginning at the one most upstream. If at any point it is found that one of these vertices belongs to the domain of dependence, then all vertices inside its domain of dependence may be added to the domain of dependence being determined and do not need to be considered further. Thus, this algorithm requires keeping track of all upstream vertices which have not yet been considered for each vertex. This algorithm continues until the domains of dependence of all vertices have been calculated.

Once these have been calculated, the domains of dependence for the control points must be calculated. The same recursion principle applies, except that it is being checked whether vertices fall within the control points' domains of dependence. Having already determined the vertex domains of dependence, if a vertex is found to fall within a control point's domain of dependence, then all vertices within that vertex's domain of dependence do not need to be considered again.

Pseudocode for both the brute-force and recursive domain of dependence algorithms is given in Appendix D.

To evaluate the performance of the recursive algorithm, these two search algorithms were compared for three different configurations: a 10° half-angle cone, a delta wing, and a straight, double-wedge wing. For each configuration, three levels of mesh refinement were used, allowing for measuring the order of complexity on each mesh. The order of complexity was estimated by fitting a power law to the computation time (averaged over five runs) over the number of vertices in the mesh. These orders of complexity were calculated over a

range of freestream Mach numbers. Since the domains of dependence become smaller with increasing Mach number, it would be anticipated that the complexity of the recursive search algorithm would increase with increasing Mach number.

The results of this study are shown in Fig. 4.4. For the cone, the order of complexity of the recursive search was a strong function of Mach number, with the order of complexity being about 1.25 at the lowest Mach number and about 1.8 at the highest Mach number. However, for the delta and double-wedge wings, the order of complexity of the recursive search remained fairly constant with Mach number. For these, the order of complexity was approximately 2 regardless of the freestream Mach number. For all configurations and Mach numbers, the order of complexity of the brute-force search was constant at about 2, as would be expected.

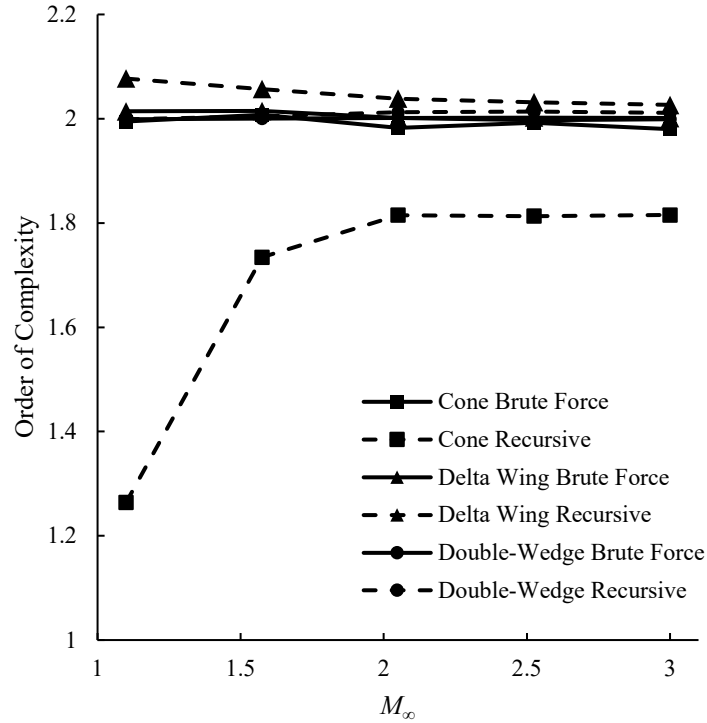


Fig. 4.4: Comparison of the orders of complexity for the brute-force and recursive domain of dependence searches for three different meshes as a function of freestream Mach number.

The reduced order of complexity for the recursive search on the cone was likely due to the slender shape of the cone. For all Mach numbers considered here, the Mach angle was greater than the cone half angle. This meant that the domain of dependence for any vertex on the cone contained a significant portion of the vertices upstream of it. The more vertices within any domain of dependence, the faster the recursive search can run.

The recursive search may have performed poorly on the delta and double-wedge wings due to the larger aspect ratio of these configurations. The aspect ratio of the delta and double-wedge wings was 1 and 4, respectively. Compare this with an aspect ratio of approximately 0.18 for the cone. For the study performed here, the configurations had zero angle of attack and sideslip. Because of this, the aspect ratio of the configuration was inversely proportional to the number of vertices contained within any one domain of dependence. The lowest Mach number considered here was 1.1, which corresponded to a Mach angle of $\mu \approx 65^\circ$. However, even with such a large Mach angle, most of the vertices for these configurations would fall outside any domain of dependence, thus degrading the performance of the recursive search.

In the limit as $M_\infty \rightarrow 1$, the Mach angle approaches 90° . Note the use of a limit is because the governing equation is not valid at $M_\infty = 1$. In this case, the Mach cone is simply a plane, and determining a domain of dependence simply amounts to determining all upstream points. In this case, the recursive search algorithm becomes extremely efficient. Thus, at very low supersonic Mach numbers, it can be anticipated that the recursive search algorithm would have reduced order of complexity on even the delta and double-wedge wings compared to the brute force search.

Unfortunately, the use of a panel method at a freestream Mach number of even 1.1 is questionable, as the flow at this Mach number is likely transonic. For extremely slender bodies, the recursive search algorithm does provide some computational savings even at high Mach numbers. However, the number of realistic configurations for which the recursive search algorithm would outperform the brute-force algorithm at Mach numbers appropriate for a panel method is limited.

There are three other aspects of the recursive search method that make it less desirable for implementation than the brute-force method. The first is that the brute-force method is very easy to implement in parallel. Calculating the domain of dependence for a given control point is independent of the domain of dependence calculations for all other control points, and so the work can easily be divided between multiple processors. Even the simplest of modern laptops are often capable of running four processes at once, which would lead to a four-fold increase in speed for the brute-force method. On the other hand, it is not currently clear that the recursive method could be implemented in parallel. All domain of dependence calculations within the recursive method are dependent upon previous calculations. It may be possible to implement the recursive search in parallel, but doing so would likely be difficult.

The second undesirable aspect of the recursive search method is that it is not built to directly calculate the domains of dependence needed for a panel method. What the recursive algorithm determines is which vertices fall inside the domains of dependence for other vertices. What is needed for a panel method is which vertices (and the panels they define) fall inside the domains of dependence for the control points. As described above, it is possible to use the vertex-vertex relationships to determine the control point domains of dependence, but this adds an extra step which could negate any speed gains from the recursive method. On the other hand, the brute-force method may be used to calculate control point domains of dependence directly. In addition, once it is known which vertices belong to which domains of dependence, it must then be determined which edges and panels partially intersect the domains of dependence. This cannot be done recursively and constitutes a significant portion of the computational expense of determining domains of dependence.

Finally, the recursive search method must be performed before any of the influence calculations, whereas the brute-force method may be done at the same time as the influence calculations (e.g. see [7, 33, 65]). Performing all domain of dependence calculations a priori requires a substantial amount of memory, as the domain of dependence relationships for all

vertices, edges, and panels relative to all control points must be stored. With such a priori storage implemented in MachLine, it was found that a mesh with approximately 50,000 panels required more than 20 GB of computer RAM to run through MachLine. This is prohibitively expensive. In addition, many of the domain of dependence calculations are also performed as part of the influence calculation procedure (e.g. Eq. (4.2) amounts to calculating R_B^2 , given by Eqs. (5.42), (5.45), or (5.48)). Thus, performing the domain of dependence calculations simultaneously with the influence calculations makes sense.

For these reasons, the brute-force method has been implemented in MachLine. With further development, it may be possible for the recursive search algorithm to be made more efficient than the brute-force approach. However, this does not seem likely at the current time.

CHAPTER 5

SOURCE AND DOUBLET PANELS

In three-dimensional irrotational flow there is nothing to correspond to the complex potential, and for this reason the representation of closed bodies, and indeed other boundaries, by suitable singularities is practised more frequently. - Bryan Thwaites [23]

Very basically, all panel methods involve breaking a configuration surface up into discrete elements (panels) and placing distributions of sources and doublets on each panel. These sources and doublets are together referred to as aerodynamic *singularities*, due to the singular nature of the flow they induce. These source and doublet distributions then perturb the overall velocity field, either throughout the flow region in subsonic flow or within the downstream Mach cone in supersonic flow. As discussed previously, by adjusting the strength of these source and doublet distributions, the flow perturbations may be made to meet boundary conditions at certain points in the flow.

To be able to evaluate the influence integrals in Eq. (1.126), the source and doublet distributions over each panel are approximated by polynomials. Typically, these distributions are specified in terms of the source and doublet strengths at certain points. The distributions may be constant, linear, quadratic, even cubic. For supersonic panel methods, quadratic-doublet-linear-source or linear-doublet-constant-source panels are typically used. Each distribution has its own unique behavior and formulas necessary for calculating its influence and surface properties.

5.1 Types of Distributions and their Characteristics

In this work, it is assumed that the doublet distribution on a panel is at most quadratic. With this, if the local-scaled coordinates of a given point on the surface of the panel are (ξ, η) , then the doublet strength at that point is given by

$$\mu(\xi, \eta) = \mu_0 + \mu_\xi \xi + \mu_\eta \eta + \frac{1}{2} \mu_{\xi\xi} \xi^2 + \mu_{\xi\eta} \xi \eta + \frac{1}{2} \mu_{\eta\eta} \eta^2 \quad (5.1)$$

where μ_0 is the doublet strength at the panel centroid, μ_ξ and μ_η are the first derivatives of doublet strength with respect to the panel coordinates, and $\mu_{\xi\xi}$, $\mu_{\xi\eta}$, and $\mu_{\eta\eta}$ are the second derivatives of doublet strength with respect to the panel coordinates. Throughout this chapter, ξ and η always represent surface coordinates in the local-scaled frame. A linear doublet distribution is obtained by dropping all quadratic terms from Eq. (5.1), resulting in

$$\mu(\xi, \eta) = \mu_0 + \mu_\xi \xi + \mu_\eta \eta \quad (5.2)$$

It is also assumed that the source distribution is at most linear, given by

$$\sigma(\xi, \eta) = \sigma_0 + \sigma_\xi \xi + \sigma_\eta \eta \quad (5.3)$$

where σ_0 is the source strength at the panel centroid and σ_ξ and σ_η are the first derivatives of source strength with respect to the panel coordinates. A constant source distribution is obtained by dropping the linear terms in Eq. (5.3), resulting in

$$\sigma(\xi, \eta) = \sigma_0 \quad (5.4)$$

Recall that the local doublet strength on the surface represents the jump in velocity potential from the inside to the outside of the configuration (see Section 1.3.3). Similarly, the local source strength represents the jump in the conormal derivative of velocity potential from the inside to the outside of the configuration. This is the same as the jump

in perturbation velocity in the conormal direction. The doublet strength also represents a jump in velocity but indirectly. If the doublet strength is not constant, then the gradient of doublet strength represents the jump in the tangential component of velocity.

For users of a panel method, the surface source and doublet strengths are not the quantities of interest. Rather, the surface pressures are typically sought, which are calculated from the surface velocity. Because of this, it is best to choose the order of source and doublet distributions such that the velocity jump due to each is of the same order. Hence, linear-doublet-constant-source or quadratic-doublet-linear-source panels are typically used. A linear-doublet-constant-source panel will have a constant jump in velocity due to both sources and doublets, and a quadratic-doublet-linear-source panel will have a linear jump in velocity due to both sources and doublets. Based on this convention, a panel method which has doublet distributions that are one degree greater than the source distributions is typically called *consistent* [50, 52, 87].

The consistency of a panel method is also determined by the geometry of the panels used. Flat panels are consistent with linear-doublet-constant-source panels, and parabolic panels are consistent with quadratic-doublet-linear-source panels [50, 52, 87, 127]. This is strongly evident from the Morino boundary condition formulation (see Section 2.1.1) where the local source strength is proportional to the inner product of the panel normal vector and the freestream velocity vector. For a flat panel, the normal vector is constant, and so the source strength should be constant. For a curved panel, the normal vector is not constant, and so a non-constant distribution of source strength is more appropriate.

Some subsonic panel methods combine curved panels with quadratic-doublet-linear-source distributions (e.g. see [45, 87]). However, the use of panel curvature makes the evaluation of influence coefficients impossible in supersonic flow [2]. Hence, all supersonic panel methods have been developed using only flat panels, regardless of the singularity distribution order [7, 33, 61, 63, 67]. This has been maintained in the current work. Despite not being consistent with the panel geometry, quadratic-doublet-linear-source distributions

over flat panels are examined in this work to determine their relative advantages and disadvantages.

It should be noted that consistency of singularity distributions and panel geometry is not required. For example, QUADPAN initially used both constant-strength doublet and source distributions to great success for subsonic flows [50, 63]. However, this does require performing some kind of interpolation or averaging on one or the other distribution to obtain usable results, thus adding another layer of approximation.

5.2 Influence Calculations

To assemble the linear system of equations used to determine the unknown doublet strengths (Eq. (2.22)), it is necessary to evaluate the influence of the source and doublet distributions over each panel on each control point. Various methods are described in the literature for doing so. It is widely acknowledged that the method first presented by Johnson and Rubbert [79] and later refined by Johnson [45] is the most efficient method for calculating higher-order influence coefficients for incompressible and compressible subsonic flows without resorting to multipole methods, which are outside the scope of this work.

The preferred method for calculating supersonic influences is not as clear. Davis [7], who recently developed a supersonic method for unstructured meshes consisting purely of subinclined panels, used the influence calculation procedure presented by Ehlers et al. [2] based on the fundamental integrals Q_1 and w_0 . Ehlers et al. also presented formulas for calculating the influence of superinclined panels, but further examples were not found of this method used elsewhere. In Appendix E of the paper by Ehlers et al. is another method for supersonic, subinclined panels matching Johnson and Rubbert's method in nomenclature [2]. It is this method that is implemented in the PAN AIR source code [65] for supersonic, subinclined panels. However, for superinclined panels, PAN AIR employs the influence calculations presented by Epton and Magnus [33]. The influence calculation methods used in other codes, such as MARCAP or ZONAIR, are not reported in the available literature [63, 66, 82].

Because of this uncertainty, significant time has been devoted as part of this work to determining the best method for computing supersonic influence coefficients. The resultant method is presented here. Throughout, the user is typically referred to one of the preceding works for the derivation of a particular equation. The derivation of any novel relations is given here. The approach taken to developing and debugging this influence calculation method is described in Appendix E.

Originally, it was intended to produce a unified approach to calculating influence calculations for subsonic, supersonic-subinclined, and supersonic-superinclined panels into a single procedure. However, it quickly became clear that doing so would slow down the method because of extra multiplicative factors, conditional statements, etc. Instead, following the example of the PAN AIR developers, each of these procedures has been developed and implemented independently.

Throughout, the coordinates of the point of integration on the panel, \mathbf{Q} , are denoted using Greek letters (i.e. $\mathbf{Q} = (\xi, \eta, \zeta)$) and those of the evaluation point, \mathbf{P} , are denoted using Roman letters (i.e. $\mathbf{P} = (x, y, z)$) with the coordinate system specified using superscripts. Different edges and vertices are denoted using subscripts. The i -th edge of a panel begins and ends at the i -th and $(i + 1)$ -th vertices, respectively.

All arctangent functions should be implemented using a two-argument arctangent (such as the Fortran intrinsic `ATAN2`). This is necessary for reproducing the proper phase of the result.

5.2.1 Influence Integrals

The perturbation velocity potential induced at a point \mathbf{P} by an arbitrary doublet distribution across a surface S is given by

$$\phi_d(\mathbf{P}) = \frac{1}{\kappa} \iint_{S \cap D_P} \mu(\mathbf{Q}) \frac{\partial}{\partial \tilde{n}} \left(\frac{1}{R_B} \right) dS \quad (5.5)$$

where R_B is given by Eq. (1.116). Equation (5.5) is not in a convenient form due to the presence of the conormal derivative. Recalling Eq. (1.107), this term may be written as

$$\frac{\partial}{\partial \tilde{n}} \left(\frac{1}{R_B} \right) = \mathbf{n}^T [B] \nabla \left(\frac{1}{R_B} \right) \quad (5.6)$$

Working in the compressible frame, this becomes

$$\frac{\partial}{\partial \tilde{n}} \left(\frac{1}{R_B} \right) = \mathbf{n}^{\mathbf{c}T} [B^c] \nabla_{(\xi, \eta, \zeta)} \left(\left[(\xi - x)^2 + \beta^2 (\eta - y) + \beta^2 (\zeta - z) \right]^{-1/2} \right) \quad (5.7)$$

Recall $\beta^2 = 1 - M_\infty^2$. Thus, β^2 is positive for subsonic flows and negative for supersonic flows. Evaluating the gradient, this may be written

$$\frac{\partial}{\partial \tilde{n}} \left(\frac{1}{R_B} \right) = \mathbf{n}^{\mathbf{c}T} [B^c] \left\{ \begin{array}{l} - \left[(\xi - x)^2 + \beta^2 (\eta - y) + \beta^2 (\zeta - z) \right]^{-3/2} (\xi - x) \\ - \beta^2 \left[(\xi - x)^2 + \beta^2 (\eta - y) + \beta^2 (\zeta - z) \right]^{-3/2} (\eta - y) \\ - \beta^2 \left[(\xi - x)^2 + \beta^2 (\eta - y) + \beta^2 (\zeta - z) \right]^{-3/2} (\zeta - z) \end{array} \right\} \quad (5.8)$$

In the compressible frame, $[B]$ has the form (see Eq. (1.71)) [33]

$$[B^c] = \begin{bmatrix} \beta^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.9)$$

Thus, Eq. (5.8) becomes

$$\frac{\partial}{\partial \tilde{n}} \left(\frac{1}{R_B} \right) = \beta^2 \mathbf{n}^{\mathbf{c}T} \left\{ \begin{array}{l} - \left[(\xi - x)^2 + \beta^2 (\eta - y) + \beta^2 (\zeta - z) \right]^{-3/2} (\xi - x) \\ - \left[(\xi - x)^2 + \beta^2 (\eta - y) + \beta^2 (\zeta - z) \right]^{-3/2} (\eta - y) \\ - \left[(\xi - x)^2 + \beta^2 (\eta - y) + \beta^2 (\zeta - z) \right]^{-3/2} (\zeta - z) \end{array} \right\} \quad (5.10)$$

or

$$\frac{\partial}{\partial \tilde{n}} \left(\frac{1}{R_B} \right) = - \frac{\beta^2 \mathbf{n}^T (\mathbf{Q} - \mathbf{P})}{R_B^3} \quad (5.11)$$

Using this, Eq. (5.5) becomes

$$\phi_d(\mathbf{P}) = \frac{\beta^2}{\kappa} \iint_{S \cap D_P} \mu(\mathbf{Q}) \frac{\mathbf{n}^T (\mathbf{P} - \mathbf{Q})}{R_B^3} dS \quad (5.12)$$

Similarly, the velocity potential induced at a point \mathbf{P} by an arbitrary source distribution across a surface is given by

$$\phi_s(\mathbf{P}) = -\frac{1}{\kappa} \iint_{S \cap D_P} \sigma \frac{1}{R_B} dS \quad (5.13)$$

As discussed previously, the influence of a point disturbance in supersonic flow is limited to the region within the Mach cone extending downstream from that point, called the domain of influence. Because of this, a panel will exert an influence on a given point only if that point falls within the domain of influence of some finite part of the panel surface. Hence the surface integrals in Eqs. (5.5) and (5.13) are taken over the intersection of S and the domain of dependence.

Calculating supersonic influences also requires the use of finite-part integrals (introduced in Section 1.3.2). Because finite-part integrals are used, the panel influences are not always smooth or even continuous in space. While this does not cause any obvious instabilities or other issues when calculating supersonic surface quantities, further investigation is needed to understand the impact of this for calculating volume properties, such as streamlines or off-body pressure signatures.

5.2.2 Fundamental Integrals

Rather than calculate Eqs. (5.5) and (5.13) as presented above, they are evaluated as sums of fundamental integrals, H and F , whose values may be calculated quickly and analytically. For a point (x, y, z) being influenced, the H integrals are given by [2, 45, 79]

$$H(M, N, K) = \iint_S \frac{(\xi - x^{ls})^{M-1} (\eta - y^{ls})^{N-1}}{R^K} d\xi d\eta \quad (5.14)$$

where [33]

$$R = \sqrt{r(\xi - x^{ls})^2 + s(\eta - y^{ls})^2 + rs(\zeta - z^{ls})^2} \quad (5.15)$$

and (ξ, η) is the point of integration, r is the panel inclination indicator calculated from Eq. (3.11), s is the flow type indicator ($s = 1$ for subsonic flow, -1 for supersonic flow), H , M , and K are specified constants, and the surface integral is taken over the surface of the panel expressed in local-scaled coordinates. Due to the scaling introduced in the local-scaled coordinate system, R and R_B are equivalent.

The F integrals, defined by [2]

$$F_i(M, N, K) = \int_{\text{edge } i} \frac{(\xi - x^{ls})^{M-1}(\eta - y^{ls})^{N-1}}{R^K} dl_i \quad (5.16)$$

are needed to calculate the H integrals. Unlike the H integrals, which are calculated for an entire panel, the F integrals are calculated for each edge of the panel.

5.2.3 Influences in Terms of Fundamental Integrals

For a flat panel with a linear distribution of source strength, the perturbation potential induced by the panel at a field point $\mathbf{P}^{ls} = (x^{ls}, y^{ls}, z^{ls})$ may be found by combining Eqs. (5.3) and (5.13), which gives

$$\phi_s = -\frac{1}{\kappa} \iint_{S \cap D_P} (\sigma_0 + \sigma_\xi \xi + \sigma_\eta \eta) \frac{1}{R_B} dS \quad (5.17)$$

In this equation, the integration is performed in global coordinates while the source distribution is expressed in local-scaled coordinates. Hence, a change of coordinates is needed. A differential element of area expressed in local-scaled coordinates, $d\xi d\eta$, is related to dS by

$$dS = J d\xi d\eta \quad (5.18)$$

where the factor J accounts for the change in integration area due to the scaled coordinate transformation and is given by [2, 7]

$$J = B\sqrt{|M^2 n_x^c{}^2 - 1|} \quad (5.19)$$

where n_x^c is the x component of the panel normal vector when expressed in the compressibility coordinate system, given by

$$n_x^c = \mathbf{n}^g \cdot \hat{\mathbf{c}}^g \quad (5.20)$$

Thus, in local-scaled coordinates, it may be written

$$\phi_s = -\frac{J}{\kappa} \iint_{S \cap D_P} (\sigma_0 + \sigma_\xi \xi + \sigma_\eta \eta) \frac{1}{R_B} d\xi d\eta \quad (5.21)$$

By adding and subtracting $\sigma_\xi x^{ls}$ and $\sigma_\eta y^{ls}$ within the parentheses, this may be written as

$$\phi_s = -\frac{J}{\kappa} \iint_{S \cap D_P} \left(\sigma_0 + \sigma_\xi \xi + \sigma_\xi x^{ls} - \sigma_\xi x^{ls} + \sigma_\eta \eta + \sigma_\eta y^{ls} - \sigma_\eta y^{ls} \right) \frac{1}{R_B} d\xi d\eta$$

$$\phi_s = -\frac{J}{\kappa} \iint_{S \cap D_P} \left(\sigma_0 + \sigma_\xi (\xi - x^{ls}) + \sigma_\xi x^{ls} + \sigma_\eta (\eta - y^{ls}) + \sigma_\eta y^{ls} \right) \frac{1}{R_B} d\xi d\eta$$

$$\phi_s = -\frac{J}{\kappa} \iint_{S \cap D_P} \left(\sigma_0 + \sigma_\xi x^{ls} + \sigma_\eta y^{ls} \right) \frac{1}{R_B} + \sigma_\xi \frac{\xi - x^{ls}}{R_B} + \sigma_\eta \frac{\eta - y^{ls}}{R_B} d\xi d\eta$$

Using Eq. (5.14), this may be expressed in terms of the fundamental H integrals, resulting in [45]

$$\phi_s = -\frac{J}{\kappa} [\sigma(x, y)H(1, 1, 1) + \sigma_x H(2, 1, 1) + \sigma_y H(1, 2, 1)] \quad (5.22)$$

where

$$\sigma(x, y) = \sigma_0 + \sigma_\xi x^{ls} + \sigma_\eta y^{ls} \quad (5.23)$$

$$\sigma_x = \sigma_\xi \quad (5.24)$$

$$\sigma_y = \sigma_\eta \quad (5.25)$$

Note this is possible because x^{ls} and y^{ls} are constant with respect to the integration. The influence of a source panel with constant strength may be obtained simply by dropping all terms dependent on derivatives of σ .

Equation (5.22) may also be written as an inner product of two vectors, as in

$$\phi_s = -\frac{J}{\kappa} \begin{Bmatrix} H(1, 1, 1) \\ H(1, 1, 1)x^{ls} + H(2, 1, 1) \\ H(1, 1, 1)y^{ls} + H(1, 2, 1) \end{Bmatrix}^T \begin{Bmatrix} \sigma_0 \\ \sigma_\xi \\ \sigma_\eta \end{Bmatrix} \quad (5.26)$$

This form is used to write the source influence in terms of the unknown source strengths discussed subsequently.

Following a similar procedure as was used for the source-induced potential, the doublet-induced potential may also be written in terms of the H integrals. When going from the compressible coordinate system (or global), as was used in Eq. (5.12), there is also a scaling of the area over which the integral is taken. However, the conormal derivative evaluated previously may also be replaced by a derivative in the z^{ls} direction. Doing so, J appears in both the numerator and denominator of the integrand and so cancels out [2]. Thus, in local-scaled coordinates, Eq. (5.12) may be written as

$$\phi_d(\mathbf{P}) = \frac{s}{\kappa} \iint_{S \cap D_P} \mu(\mathbf{Q}) \frac{z^{ls}}{R_B^3} d\xi d\eta \quad (5.27)$$

Note that the flow type indicator, s , appears because the shift to local-scaled coordinates removes the scaling factor but does not change the character of the flow.

Now using the H integrals defined previously, for a flat panel with a quadratic distribution of doublet strength, the perturbation potential induced by the panel at a field point $\mathbf{P} = (x^{ls}, y^{ls}, z^{ls})$ is given by [45]

$$\begin{aligned} \phi_d = \frac{sh}{\kappa} & \left[\mu(x, y)H(1, 1, 3) + \mu_x(x, y)H(2, 1, 3) + \mu_y(x, y)H(1, 2, 3) + \frac{1}{2}\mu_{xx}H(3, 1, 3) \right. \\ & \left. + \mu_{xy}H(2, 2, 3) + \frac{1}{2}\mu_{yy}H(1, 3, 3) \right] \end{aligned} \quad (5.28)$$

where

$$h = z^{ls} \quad (5.29)$$

$$\mu(x, y) = \mu_0 + \mu_\xi x^{ls} + \mu_\eta y^{ls} + \frac{1}{2}\mu_{\xi\xi}x^{ls2} + \mu_{\xi\eta}x^{ls}y^{ls} + \frac{1}{2}\mu_{\eta\eta}y^{ls2} \quad (5.30)$$

$$\mu_x = \mu_\xi + \mu_{\xi\xi}x^{ls} + \mu_{\xi\eta}y^{ls} \quad (5.31)$$

$$\mu_y = \mu_\eta + \mu_{\xi\eta}x^{ls} + \mu_{\eta\eta}y^{ls} \quad (5.32)$$

$$\mu_{xx} = \mu_{\xi\xi} \quad (5.33)$$

$$\mu_{xy} = \mu_{\xi\eta} \quad (5.34)$$

$$\mu_{yy} = \mu_{\eta\eta} \quad (5.35)$$

The influence of a doublet panel with linear strength may be determined simply by dropping the quadratic terms from the above equations.

Similar to the source influence, Eq. (5.28) may be written as an inner product of two vectors, as in

$$\phi_d = \frac{sh}{\kappa} \left\{ \begin{array}{c} H(1, 1, 3) \\ H(2, 1, 3) + H(1, 1, 3)x^{ls} \\ H(1, 2, 3) + H(1, 1, 3)y^{ls} \\ \frac{1}{2}H(3, 1, 3) + H(2, 1, 3)x^{ls} + \frac{1}{2}H(1, 1, 3)x^{ls^2} \\ H(2, 2, 3) + H(2, 1, 3)y^{ls} + H(1, 2, 3)x^{ls} + H(1, 1, 3)x^{ls}y^{ls} \\ \frac{1}{2}H(1, 3, 3) + H(1, 2, 3)y^{ls} + \frac{1}{2}H(1, 1, 3)y^{ls^2} \end{array} \right\}^T \left\{ \begin{array}{c} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \\ \mu_{\eta\eta} \end{array} \right\} \quad (5.36)$$

Thus, to calculate both source- and doublet-induced potentials, one must simply calculate the required H integrals (which requires calculating some F integrals as well) and combine them with the parameters that describe the singularity distributions over each panel (σ_0 , μ_0 , etc.).

5.2.4 Geometric Quantities

To calculate the H and F integrals, several geometric quantities relating the panel and the evaluation point \mathbf{P} are needed. How each of these is calculated is dependent upon the type of flow and the inclination of the panel

One quantity that is the same for all flows and panels is the panel edge outward normal vector. The edge outward normal vector is given by

$$\hat{\mathbf{n}}^{ls} = \begin{bmatrix} \hat{t}_{y^{ls}} \\ -\hat{t}_{x^{ls}} \end{bmatrix} \quad (5.37)$$

where $\hat{\mathbf{t}}$ is the unit tangent vector for the edge. Here, the tangent vectors are assumed to proceed in the counter-clockwise direction about the panel normal vector (i.e. the tangent

vector for edge i points from vertex i to vertex $i + 1$). This is important to take into account when calculating the influence of mirrored panels. Supersonic influence calculations are particularly sensitive to this distinction. Both the tangent and normal vectors should be calculated using the vertex locations in the local-scaled system and normalized in the local-scaled system, rather than being transformed from their counterparts in the global system.

The geometric quantities for each type of flow and panel inclination are described in the following three sections. The reader may notice similarities between the subsonic geometric calculations and the supersonic, superinclined calculations. This has to do with the nature of the governing PDE, which is elliptic in subsonic flow and hyperbolic in supersonic flow. For superinclined panels, the coordinate system is rotated so that the wave direction is normal to the panel surface. This means that any in-plane geometric calculations may be performed ignorant of the wave direction. This is the same for subsonic flow because there is no wave direction in subsonic flow. However, for subinclined, supersonic panels, the wave direction is in the plane of the panel, and so the geometric calculations are heavily dependent upon wave direction. As such, the influence calculations for supersonic, subinclined panels are far more involved than the other two cases.

For all geometric quantities, the subscripts i and $i + 1$ denote the edge index. The subscripts 1 and 2 denote the start and end vertices of each edge, respectively.

Panel Edge Inclination

For subinclined panels in supersonic flow, the inclination of each edge is calculated based on the angle it makes with the freestream. If an edge makes an angle with the freestream greater than the Mach angle, it is referred to as a *supersonic* edge. If it makes an angle with the freestream less than the Mach angle, it is a *subsonic* edge. If the angle is equal to the Mach angle, it is a *sonic* edge. The inclination of each edge may be determined from the parameter

$$b = \hat{n}_{x^{ls}}^2 - \hat{n}_{y^{ls}}^2 \quad (5.38)$$

where $\hat{n}_{x^{ls}}$ and $\hat{n}_{y^{ls}}$ are the x^{ls} and y^{ls} components of the edge outward normal vector. An edge is supersonic if $b > 0$, sonic if $b = 0$, and subsonic if $b < 0$. A supersonic edge (similar to a superinclined panel) falls outside of the Mach cone emanating from any point on its length. A subsonic edge does fall within its own Mach cones.

Based on this interpretation, it may be seen that, in subsonic flows, all edges are subsonic. Similarly, for superinclined panels, all edges are supersonic. Because of this, b is only calculated for supersonic, subinclined panels. This fits with the discussion in the previous section regarding the wave direction.

Subsonic Geometric Quantities

In calculating the influence coefficients, five quantities must first be calculated: h , a , g , l , and R . The height above the panel, h , is simply the z^{ls} coordinate of \mathbf{P} (see Eq. (1.25)). This is not the true perpendicular height above the panel but is correct due to the change of coordinate system and replacing the conormal derivative with the z^{ls} derivative.

The perpendicular in-plane distance, a , from the evaluation point to the edge is given by

$$a = \hat{\mathbf{n}}^{ls} \cdot [\xi^{ls} - x^{ls}, \eta^{ls} - y^{ls}]^T \quad (5.39)$$

This quantity is constant for a given edge, and so may be calculated using any point (ξ^{ls}, η^{ls}) on the edge. Within MachLine, it is calculated using vertex i for the i -th edge. The perpendicular distance to the edge, g , is also constant for an edge and is given by

$$g_i^2 = a^2 + h^2 \quad (5.40)$$

In the calculation of H and F , only g_i^2 and never g is used, so the square root does not need to be computed. The in-plane tangential distance along an edge, l_i , is given by

$$l = \hat{\mathbf{t}} \cdot [\xi^{ls} - x^{ls}, \eta^{ls} - y^{ls}]^T = -\hat{n}_{y^{ls}}(\xi^{ls} - x^{ls}) + \hat{n}_{x^{ls}}(\eta^{ls} - y^{ls}) \quad (5.41)$$

It is defined for each endpoint of an edge (i.e. for the i -th edge, both l_{1_i} and l_{2_i} must be calculated). Lastly, the radial distance, R , from each endpoint is given by

$$R^2 = g^2 + l^2 = (\xi^{ls} - x^{ls})^2 + (\eta^{ls} - y^{ls})^2 + h^2 \quad (5.42)$$

Either form of Eq. (5.42) may be used in the subsonic case. As with l , both R_1 and R_2 are calculated for each edge. Some computational savings may be gained by recognizing that $R_{2_i} = R_{1_{i+1}}$.

Supersonic, Subinclined Geometric Quantities

For subinclined panels, h and a are defined as in the subsonic case. The perpendicular distance, g^2 , is then calculated using

$$g^2 = a^2 - bh^2 \quad (5.43)$$

where b is given by Eq. (5.38). Note that, unlike the subsonic case, g^2 may be negative for a given edge. If this is the case, then the point under consideration is outside the domain of dependence of the edge. The tangential distance, l , is given by

$$l = \hat{n}_{y^{ls}}(\xi^{ls} - x^{ls}) + \hat{n}_{x^{ls}}(\eta^{ls} - y^{ls}) \quad (5.44)$$

and the hyperbolic distance from edge endpoint is given by

$$R^2 = (\xi^{ls} - x^{ls})^2 - (\eta^{ls} - y^{ls})^2 - h^2 \quad (5.45)$$

This formula for R is used in MachLine instead of Eq. (E15) in [2] as the latter quickly loses precision when b is close to zero.

As with g^2 , R^2 may be negative. If this is the case, then the point under consideration falls outside the Mach cone emanating from the edge endpoint. If R^2 is negative, then it is set to zero. If $R_1 = 0$, then

$$l_{1i} = -\sqrt{|g_i^2|} \quad (5.46)$$

and if $R_2 = 0$, then

$$l_{2i} = \sqrt{|g_i^2|} \quad (5.47)$$

This is done so that the formulas used to calculate the H and F integrals remain valid even when an edge intersects the Mach cone [7, 65]. Eqs. (5.46)-(5.47) may be derived from Eq. (E15) in [2] by setting $R^2 = 0$. Note that if l_{1i} and l_{2i} are used to calculate domains of dependence, then the correction given by Eqs. (5.46)-(5.47) must be performed only after the domains of dependence have been determined.

Supersonic, Superinclined Geometric Quantities

To promote unity with the other influence calculation procedures, the conventions presented by Ehlers et al. [2] and Epton and Magnus [33] have been modified slightly here. Here, for superinclined panels, the z^{ls} direction is normal to the plane of the panel, and x^{ls} and y^{ls} lie in the plane of the panel. This is consistent with the coordinate transformation defined in Appendix 3.2.

With this convention, the quantities h , a , and l are defined as in the subsonic case. The only calculation which differs is the hyperbolic distance from each panel corner, given by

$$R^2 = -(\xi^{ls} - x^{ls})^2 - (\eta^{ls} - y^{ls})^2 + h^2 \quad (5.48)$$

As with the subinclined case, if R^2 is negative, then that vertex falls outside the domain of dependence and R is set to zero. If then $R_1 = 0$, then

$$l_{1i} = -1 \quad (5.49)$$

and if $R_2 = 0$, then

$$l_{2_i} = 1 \quad (5.50)$$

Again, this is done so that the formulas for the fundamental integrals remain valid when the edge intersects the Mach cone. The edge perpendicular distance, g^2 , is not used for the supersonic, superinclined influence calculations.

5.2.5 Calculation of H Integrals

To reduce computation times, the necessary H integrals are calculated using a series of recursion relations, rather than each being calculated explicitly [2, 45, 79]. Johnson and Rubbert [79] first presented this method for subsonic flows. It was later refined for subsonic flows by Johnson [45] and for subinclined panels in supersonic flows by Ehlers et al. [2]. The method used in PAN AIR [33] is essentially equivalent. Using different nomenclature, Epton and Magnus [33] applied this to superinclined panels. Their method has been adapted here to use nomenclature consistent with the other methods. For this method of influence calculation, the integral $hH(1, 1, 3)$ is calculated directly, and all other necessary integrals are obtained using the recursion formulas. These recursion formulas are derived in Appendix H.

Subsonic Formulas

From Johnson [45], $hH(1, 1, 3)$ is given by

$$hH(1, 1, 3) = \text{sign}(h) \sum_i \arctan \left(\frac{a_i(l_{2_i}c_{1_i} - l_{1_i}c_{2_i})}{c_{1_i}c_{2_i} + a_i^2 l_{1_i}l_{2_i}} \right) \quad (5.51)$$

where $c = g_i^2 + |h|\sqrt{l^2 + g_i^2} = g_i^2 + |h|R$.

Using Eq. (G.44) from [45], $H(1, 1, 1)$ is then given by

$$H(1, 1, 1) = -h^2 H(1, 1, 3) + \sum_i a_i F_i(1, 1, 1) \quad (5.52)$$

The calculation of $F_i(1, 1, 1)$ is discussed in the following section.

In the method presented by Johnson [45], the exact formulas for calculating $H(1, 1, 1)$ and $hH(1, 1, 3)$ are dependent upon the magnitude of h . This is because one of the recursion relations used by Johnson involves dividing by h , which may be very small (the particular formula in question is (D.42) in [45] which calculates $H(1, 1, K)$ from $H(1, 1, K - 2)$). To avoid potential numerical issues in such cases, Johnson reverses the recursion when h is very small so that h is in the numerator. However, Eq. (D.42) in [45] is only used once when calculating potential influences for flat panels. In this case, it is in the form of Eq. (5.52), which does not require dividing by h . Hence the above method of calculating the H integrals is stable regardless of the magnitude of h .

From Eqs. (G.42) and (G.43) in [45]

$$H(2, 1, 3) = - \sum_i \hat{n}_{\xi_i}^{ls} F_i(1, 1, 1) \quad (5.53)$$

$$H(1, 2, 3) = - \sum_i \hat{n}_{\eta_i}^{ls} F_i(1, 1, 1) \quad (5.54)$$

For calculating the influence of a linear source distribution, $H(2, 1, 1)$ and $H(1, 2, 1)$ are also necessary. These are given by [45]

$$H(2, 1, 1) = \frac{1}{2} \left[-h^2 H(2, 1, 3) + \sum_i a_i F_i(2, 1, 1) \right] \quad (5.55)$$

$$H(1, 2, 1) = \frac{1}{2} \left[-h^2 H(1, 2, 3) + \sum_i a_i F_i(1, 2, 1) \right] \quad (5.56)$$

respectively.

For a quadratic doublet distribution, the following integrals must also be calculated [45]

$$H(3, 1, 3) = H(1, 1, 1) - \sum_i \hat{n}_{\xi_i}^{ls} F_i(2, 1, 1) \quad (5.57)$$

$$H(2, 2, 3) = - \sum_i \hat{n}_{\xi_i}^{ls} F_i(1, 2, 1) \quad (5.58)$$

$$H(1, 3, 3) = H(1, 1, 1) - \sum_i \hat{n}_{\eta_i}^{ls} F_i(1, 2, 1) \quad (5.59)$$

During implementation, it is useful to be able to check that these integrals are correctly calculated. This may be done using the additional relations [2, 45]

$$H(2, 2, 3) = - \sum_i \hat{n}_{\xi_i}^{ls} F_i(1, 2, 1) = - \sum_i \hat{n}_{\eta_i}^{ls} F_i(2, 1, 1) \quad (5.60)$$

$$H(1, 1, 1) = H(3, 1, 3) + H(1, 3, 3) + h^2 H(1, 1, 3) \quad (5.61)$$

Supersonic, Subinclined Formulas

From Ehlers et al. [2], $hH(1, 1, 3)$ is calculated again by a sum over each edge of the panel

$$hH(1, 1, 3) = \sum hH_i(1, 1, 3) \quad (5.62)$$

If the point under consideration falls in the Mach wedge region of edge i (i.e. the edge intersects the domain of dependence but $R_{1_i} = R_{2_i} = 0$), then

$$hH_i(1, 1, 3) = \text{sign}(h\hat{n}_{\xi_i}^{ls})\pi \quad (5.63)$$

Otherwise,

$$hH_i(1, 1, 3) = \arctan \left(\frac{ha_i F_1}{R_{1_i} R_{2_i} + h^2 F_2} \right) \quad (5.64)$$

where, for each edge,

$$F_1 = \begin{cases} \frac{l_{1_i} R_{2_i} - l_{2_i} R_{1_i}}{g_i^2}, & b_i > 0 \\ \frac{(R_{2_i} - R_{1_i})(R_{2_i} + R_{1_i})}{l_{1_i} R_{2_i} + l_{2_i} R_{1_i}}, & b_i \leq 0 \end{cases} \quad (5.65)$$

$$F_2 = \begin{cases} \frac{b_i R_{1_i} R_{2_i} + l_{1_i} l_{2_i}}{g_i^2}, & b_i > 0 \\ \frac{g_i^2 - l_{1_i}^2 - l_{2_i}^2}{b_i R_{1_i} R_{2_i} - l_{1_i} l_{2_i}}, & b_i \leq 0 \end{cases} \quad (5.66)$$

Of course, if the edge falls entirely outside the domain of dependence, then $hH_i(1, 1, 3) = 0$.

Following the PAN AIR source code [65] and Ehlers et al. [2], the rest of the lower-order integrals are given by

$$H(1, 1, 1) = h^2 H(1, 1, 3) + \sum_i a_i F_i(1, 1, 1) \quad (5.67)$$

$$H(2, 1, 3) = - \sum_i \hat{n}_{\xi_i}^{ls} F_i(1, 1, 1) \quad (5.68)$$

$$H(1, 2, 3) = \sum_i \hat{n}_{\eta_i}^{ls} F_i(1, 1, 1) \quad (5.69)$$

For a linear source distribution, it is also necessary to calculate

$$H(2, 1, 1) = \frac{1}{2} \left[h^2 H(2, 1, 3) + \sum_i a_i F_i(2, 1, 1) \right] \quad (5.70)$$

$$H(1, 2, 1) = \frac{1}{2} \left[h^2 H(1, 2, 3) + \sum_i a_i F_i(1, 2, 1) \right] \quad (5.71)$$

and for a quadratic doublet distribution, it is necessary to calculate [2]

$$H(3, 1, 3) = H(1, 1, 1) - \sum_i \hat{n}_{\xi_i}^{ls} F_i(2, 1, 1) \quad (5.72)$$

$$H(2, 2, 3) = - \sum_i \hat{n}_{\xi_i}^{ls} F_i(1, 2, 1) \quad (5.73)$$

$$H(1, 3, 3) = -H(1, 1, 1) + \sum_i \hat{n}_{\eta_i}^{ls} F_i(1, 2, 1) \quad (5.74)$$

Again, the implementation of these integral calculations may be checked using

$$H(2, 2, 3) = - \sum_i \hat{n}_{\xi_i}^{ls} F_i(1, 2, 1) = \sum_i \hat{n}_{\eta_i}^{ls} F_i(2, 1, 1) \quad (5.75)$$

$$H(1, 1, 1) = H(3, 1, 3) - H(1, 3, 3) - h^2 H(1, 1, 3) \quad (5.76)$$

Supersonic, Superinclined Formulas

In the superinclined case, the calculation of $hH(1, 1, 3)$ is rather different than the other two cases. One reason for this is that the domain of dependence may intersect the panel surface without intersecting any of the panel edges. Because of this, $hH(1, 1, 3)$ is first initialized to 2π . Then, for each edge of the panel which intersects the domain of dependence, $hH(1, 1, 3)$ is decremented by π . For each corner of the panel which falls inside the domain of dependence, π is added to $hH(1, 1, 3)$, thus cancelling out the edge influence just added. Then, for every corner i of the panel which falls inside the domain of dependence, $hH(1, 1, 3)$ is updated by

$$hH(1, 1, 3) = hH(1, 1, 3) - \arctan(Y, -X) \quad (5.77)$$

where for each corner (or vertex) i

$$X = a_i a_{i-1} - h^2 \hat{\mathbf{t}}_i^{ls} \cdot \hat{\mathbf{t}}_{i-1}^{ls} \quad (5.78)$$

$$Y = hR_{1_i} \left(\hat{\mathbf{t}}_{i-1}^{ls} \times \hat{\mathbf{t}}_i^{ls} \right)_{\zeta^{ls}} \quad (5.79)$$

Thus, a concise formula for $hH(1, 1, 3)$ (c.f. Eq. (J.8.109) in [33]) is

$$hH(1, 1, 3) = 2\pi - \sum_{\text{edges}} \pi + \sum_{\text{corners}} \pi + \sum_{\text{corners}} \arctan(Y, -X) \quad (5.80)$$

The other lower-order integrals are then given by

$$H(1, 1, 1) = -h^2 H(1, 1, 3) + \sum_i a_i F_i(1, 1, 1) \quad (5.81)$$

$$H(2, 1, 3) = \sum_i \hat{n}_{\xi_i}^{ls} F_i(1, 1, 1) \quad (5.82)$$

$$H(1, 2, 3) = \sum_i \hat{n}_{\eta_i}^{ls} F_i(1, 1, 1) \quad (5.83)$$

For a linear source distribution, it is also necessary to calculate

$$H(2, 1, 1) = \frac{1}{2} \left[-h^2 H(2, 1, 3) + \sum_i a_i F_i(2, 1, 1) \right] \quad (5.84)$$

$$H(1, 2, 1) = \frac{1}{2} \left[-h^2 H(1, 2, 3) + \sum_i a_i F_i(1, 2, 1) \right] \quad (5.85)$$

For a quadratic doublet distribution, it is also necessary to calculate

$$H(3, 1, 3) = -H(1, 1, 1) + \sum_i \hat{n}_{\xi_i}^{ls} F_i(2, 1, 1) \quad (5.86)$$

$$H(2, 2, 3) = \sum_i \hat{n}_{\xi_i}^{ls} F_i(1, 2, 1) \quad (5.87)$$

$$H(1, 3, 3) = -H(1, 1, 1) + \sum_i \hat{n}_{\eta_i}^{ls} F_i(2, 1, 1) \quad (5.88)$$

The implementation of these calculations may be checked using

$$H(2, 2, 3) = \sum_i \hat{n}_{\xi_i}^{ls} F_i(1, 2, 1) = \sum_i \hat{n}_{\eta_i}^{ls} F_i(2, 1, 1) \quad (5.89)$$

$$H(1, 1, 1) = -H(3, 1, 3) - H(1, 3, 3) - h^2 H(1, 1, 3) \quad (5.90)$$

5.2.6 Calculation of F Integrals

The F integrals are necessary for calculating the H integrals using the above recursion relations. The F integrals are calculated for each edge. As with the panel integrals, different formulas are used based on the flow (subsonic or supersonic) and the panel inclination.

For linear-doublet-constant-source panels, only $F_i(1, 1, 1)$ is needed for each edge. For quadratic-doublet-linear-source panels, $F_i(2, 1, 1)$ and $F_i(1, 2, 1)$ are also needed.

Subsonic Formulas

Johnson [45] gives recursion relations for calculating all the F integrals based on the value of $F_i(1, 1, 1)$, which is given by

$$F_i(1, 1, 1) = \begin{cases} \ln \left(\frac{(\sqrt{l_{1_i}^2 + g_i^2} - l_{1_i})(\sqrt{l_{2_i}^2 + g_i^2} + l_{2_i})}{g_i^2} \right), & \text{sign}(l_{1_i}) \neq \text{sign}(l_{2_i}) \\ \text{sign}(l_{1_i}) \ln \left(\frac{\sqrt{l_{2_i}^2 + g_i^2} + |l_{2_i}|}{\sqrt{l_{1_i}^2 + g_i^2} + |l_{1_i}|} \right), & \text{otherwise} \end{cases} \quad (5.91)$$

This is modified slightly from Eq. (D.60) in [45]; however, the two are equivalent. Proof of this is given in Appendix G.

Because $E_i(1, 1, -1) = R_{2_i} - R_{1_i}$ (see [45] for the definition of $E()$), the formulas given by Johnson [45] for $F_i(2, 1, 1)$ and $F_i(1, 2, 1)$ reduce to

$$F_i(2, 1, 1) = a_i \hat{n}_{\xi_i}^{ls} F_i(1, 1, 1) - \hat{n}_{\eta_i}^{ls} (R_{2_i} - R_{1_i}) \quad (5.92)$$

$$F_i(1, 2, 1) = a_i \hat{n}_{\eta_i}^{ls} F_i(1, 1, 1) + \hat{n}_{\xi_i}^{ls} (R_{2_i} - R_{1_i}) \quad (5.93)$$

Supersonic, Subinclined Formulas

The supersonic, subinclined formulas for the necessary F integrals are given by Ehlers et al. [2] and reproduced within the PAN AIR source code. Their calculation is dependent upon whether the edge is supersonic, subsonic, or *nearly-sonic*, meaning the edge is closely aligned with the Mach cone [2]. The simplest case is when the edge is supersonic (i.e. $b_i > 0$) and the point under consideration falls within the Mach wedge. In this case

$$F_i(1, 1, 1) = \frac{\pi}{\sqrt{|b_i|}} \quad (5.94)$$

and

$$F_i(2, 1, 1) = a_i \hat{n}_{\xi_i}^{ls} F_i(1, 1, 1) / b_i \quad (5.95)$$

$$F_i(1, 2, 1) = -a_i \hat{n}_{\eta_i}^{ls} F_i(1, 1, 1) / b_i \quad (5.96)$$

When an edge is exactly aligned with the Mach cone, the standard formulas for $F_i(M, N, K)$ become impossible to evaluate directly due to b being zero [2, 7]. To avoid this, a series expansion is used for nearly-sonic edges. An edge is considered nearly-sonic if

$$|F_2| > 125 \left| \sqrt{|b_i|} F_1 \right| \quad (5.97)$$

where F_1 and F_2 are given by Eqs. (5.65) and (5.66). If this is the case, then $F_i(1, 1, 1)$ is given by [2]

$$F_i(1, 1, 1) = -\varepsilon + b_i S \quad (5.98)$$

where

$$\varepsilon = \frac{F_1}{F_2} \quad (5.99)$$

$$S = \varepsilon^3 \left[\frac{1}{3} - \frac{b_i \varepsilon^2}{5} + \frac{b_i^2 \varepsilon^4}{7} \right] \quad (5.100)$$

and F_1 and F_2 are given by Eqs. (5.65) and (5.66), respectively.

For a nearly-sonic edge, the higher-order integrals are given by [2]

$$F_i(1, 2, 1) = \left[-\hat{n}_{\xi_i}^{ls} (R_{2_i} - R_{1_i}) R_{1_i} R_{2_i} + l_{2_i} R_{1_i} (\eta_{1_i}^{ls} - y^{ls}) - l_{1_i} R_{2_i} (\eta_{2_i}^{ls} - y^{ls}) \right] / (g_i^2 F_2) - a_i \hat{n}_{\eta_i}^{ls} S \quad (5.101)$$

$$F_i(2, 1, 1) = -\hat{n}_{\eta_i}^{ls} (R_{2_i} - R_{1_i}) + a_i \hat{n}_{\xi_i}^{ls} F_i(1, 1, 1) - 2 \hat{n}_{\xi_i}^{ls} \hat{n}_{\eta_i}^{ls} F_i(1, 2, 1) \quad (5.102)$$

These formulas have been modified from the formulas given in [2] based on the PAN AIR source code [65]. The modifications are for computational efficiency.

The condition for a nearly-sonic edge was modified somewhat from that used in the PAN AIR source code [65]. Ehlers et al. simply recommend the nearly-sonic calculation be used when $F_2 \gg \sqrt{|b|} |F_1|$ (see Eq. (E23) in [2]). In the PAN AIR source code, the constant in Eq. (5.97) was 100, and this was initially copied in MachLine. However, it was discovered that this caused issues with the calculation of $F_i(1, 2, 1)$ when g^2 was small. Increasing the constant in Eq. (5.97) alleviated these issues; hence the recommended value of 125. Increasing this value amounted to reducing the window for where an edge would be considered nearly-sonic.

If the edge is subsonic, then [2]

$$F_i(1, 1, 1) = -\frac{\text{sign}(\hat{n}_{\eta_i}^{ls})}{\sqrt{|b|}} \log \left(\frac{\sqrt{|b|} R_{1_i} + |l_{1_i}|}{\sqrt{|b|} R_{2_i} + |l_{2_i}|} \right) \quad (5.103)$$

If the edge is supersonic (excepting the Mach-wedge case), $F_i(1, 1, 1)$ is given by

$$F_i(1, 1, 1) = -\frac{1}{\sqrt{b}} \arctan \left(\frac{\sqrt{b} F_1}{F_2} \right) \quad (5.104)$$

For both subsonic and supersonic edges (again excepting the Mach-wedge condition), $F_i(2, 1, 1)$ and $F_i(1, 2, 1)$ are given by [2]

$$F_i(1, 2, 1) = - \left[\hat{n}_{\xi_i}^{ls} (R_{2_i} - R_{1_i}) + a \hat{n}_{\eta_i}^{ls} F_i(1, 1, 1) \right] / b \quad (5.105)$$

$$F_i(2, 1, 1) = -\hat{n}_{\eta_i}^{ls} (R_{2_i} - R_{1_i}) + a \hat{n}_{\xi_i}^{ls} F_i(1, 1, 1) - 2 \hat{n}_{\xi_i}^{ls} \hat{n}_{\eta_i}^{ls} F_i(1, 2, 1) \quad (5.106)$$

As with the H integrals, the implementation of the higher-order F integral calculations may be verified using the relation

$$\hat{n}_{\xi_i}^{ls} F_i(2, 1, 1) + \hat{n}_{\eta_i}^{ls} F_i(1, 2, 1) = a F_i(1, 1, 1) \quad (5.107)$$

Supersonic, Superinclined Formulas

All edges of a superinclined panel are supersonic, and so the Mach wedge condition is always a possibility. If $R_{1_i} = R_{2_i} = 0$, then

$$F_i(1, 1, 1) = -\pi \quad (5.108)$$

Otherwise,

$$F_i(1, 1, 1) = \arctan \left(\frac{F_1}{F_2} \right) \quad (5.109)$$

where

$$F_1 = l_{1_i} R_{2_i} - l_{2_i} R_{1_i} \quad (5.110)$$

$$F_2 = R_{1_i} R_{2_i} + l_{1_i} l_{2_i} \quad (5.111)$$

In all cases, $F_i(2, 1, 1)$ and $F_i(1, 2, 1)$ are given by

$$F_i(2, 1, 1) = a_i \hat{n}_{\xi_i}^{ls} F_i(1, 1, 1) - \hat{n}_{\eta_i}^{ls} (R_{2_i} - R_{1_i}) \quad (5.112)$$

$$F_i(1, 2, 1) = a_i \hat{n}_{\eta_i}^{ls} F_i(1, 1, 1) + \hat{n}_{\xi_i}^{ls} (R_{2_i} - R_{1_i}) \quad (5.113)$$

For the Mach-wedge condition, $R_{2_i} - R_{1_i}$ is identically zero.

The implementation of these equations may be verified using

$$\hat{n}_{\xi_i}^{ls} F_i(2, 1, 1) + \hat{n}_{\eta_i}^{ls} F_i(1, 2, 1) = a_i F_i(1, 1, 1) \quad (5.114)$$

5.3 Strength and Parameter Spaces

The vectors of variables used in Eqs (5.26) and (5.36), i.e.

$$\{\sigma\} = \begin{Bmatrix} \sigma_0 \\ \sigma_\xi \\ \sigma_\eta \end{Bmatrix} \quad (5.115)$$

$$\{\mu\} = \begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \\ \mu_{\eta\eta} \end{Bmatrix} \quad (5.116)$$

are referred to here as the *source and doublet distribution parameters*. These vectors are said to exist in *parameter space*. The velocity potentials induced by source and doublet distribution parameters are most easily expressed in this parameter form

However, a panel method is typically formulated with the source and doublet *strengths* at certain points as the unknowns to the problem, rather than these parameters. The vectors

of source and doublet strengths influencing a given panel's distributions are denoted $\{S\}$ and $\{M\}$, respectively. These vectors are said to exist in *strength space*. In order to assemble a linear system of equations in terms of the singularity strengths, a transformation from strength to parameter space is needed. Once this transformation is determined (usually a matrix), Eqs. (5.26) and (5.36) may be written as

$$\phi_s = -\frac{J}{\kappa} \left\{ \begin{array}{c} H(1, 1, 1) \\ H(1, 1, 1)x^{ls} + H(2, 1, 1) \\ H(1, 1, 1)y^{ls} + H(1, 2, 1) \end{array} \right\}^T [T_\sigma]\{S\} \quad (5.117)$$

$$\phi_d = \frac{sh}{\kappa} \left\{ \begin{array}{c} H(1, 1, 3) \\ H(2, 1, 3) + H(1, 1, 3)x^{ls} \\ H(1, 2, 3) + H(1, 1, 3)y^{ls} \\ \frac{1}{2}H(3, 1, 3) + H(2, 1, 3)x^{ls} + \frac{1}{2}H(1, 1, 3)x^{ls^2} \\ H(2, 2, 3) + H(2, 1, 3)y^{ls} + H(1, 2, 3)x^{ls} + H(1, 1, 3)x^{ls}y^{ls} \\ \frac{1}{2}H(1, 3, 3) + H(1, 2, 3)y^{ls} + \frac{1}{2}H(1, 1, 3)y^{ls^2} \end{array} \right\}^T [T_\mu]\{M\} \quad (5.118)$$

where $[T_\sigma]$ and $[T_\mu]$ are the strength-parameter transformation matrices, defined by

$$\{\sigma\} = [T_\sigma]\{S\} \quad (5.119)$$

$$\{\mu\} = [T_\mu]\{M\} \quad (5.120)$$

The strength-parameter transformations are dependent upon whether lower-order (i.e. linear-doublet-constant-source) or higher-order (i.e. quadratic-doublet-linear-source) panels are used. For lower-order panels, the lower-order transformation is relatively straightforward. The implementation described here was adapted from Davis [7, 68]. The latter case has only been considered for structured meshes. For example, the developers of PAN AIR used spline fits of doublet and source strengths at neighboring panel centroids to determine

the source and doublet parameters for a given panel [2, 33]. Various methods for using higher-order panels on an unstructured mesh are examined here.

5.4 Transformations for Lower-Order Distributions

Here, lower-order distributions refers to panels with linear doublet and constant source distributions. This is a departure from the literature, in which the term “lower-order” typically denotes constant-strength doublet panels [80].

5.4.1 Linear Doublet Distribution

In the current work, the doublet strengths are defined at each panel vertex. For triangular panels, these vertex-associated strengths then uniquely define a linear distribution of doublet strength on the panel surface. In addition, the doublet strength is exactly continuous between panels because each edge will have the same linear distribution for both panels touching it.

For a linear doublet distribution, both the parameter and strength spaces are three-dimensional. The strength vector $\{M\}$ consists of the doublet strengths defined at the three corners of the panel. The matrix equation relating doublet strengths to distribution parameters is obtained by writing Eq. (5.2) for each panel vertex, resulting in

$$\begin{bmatrix} 1 & \xi_1 & \eta_1 \\ 1 & \xi_2 & \eta_2 \\ 1 & \xi_3 & \eta_3 \end{bmatrix} \begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \end{Bmatrix} = \begin{Bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{Bmatrix} \quad (5.121)$$

where (ξ_i, η_i) is the location of the i -th vertex in local-scaled coordinates, and μ_i is the doublet strength at the i -th vertex. The matrix in this equation is easily inverted to yield the necessary transformation from strength space to parameter space. This gives

$$\{\mu\} = \left(\begin{bmatrix} 1 & \xi_1 & \eta_1 \\ 1 & \xi_2 & \eta_2 \\ 1 & \xi_3 & \eta_3 \end{bmatrix} \right)^{-1} \{M\} \quad (5.122)$$

To simplify notation, a matrix is introduced called $[S_\mu]$. Regardless of the dimension of the strength and parameter spaces, $[S_\mu]$ is defined as the matrix which satisfies

$$[S_\mu]\{\mu\} = \{M\} \quad (5.123)$$

The matrix $[S_\mu]$ may be obtained by writing Eq. (5.1) for each point represented in $\{M\}$ including only the distribution parameters present in $\{\mu\}$ (i.e. for a linear distribution, the $\mu_{\xi\xi}$, $\mu_{\xi\eta}$, and $\mu_{\eta\eta}$ terms are dropped). Thus

$$[T_\mu] = [S_\mu]^{-1} \quad (5.124)$$

for linear doublet distributions.

5.4.2 Constant Source Distribution

A constant source distribution requires only one distribution parameter, σ_0 . Since the source strength is defined at each panel centroid, the distribution parameters are

$$\sigma_0 = \sigma_i \quad (5.125)$$

$$\sigma_\xi = 0 \quad (5.126)$$

$$\sigma_\eta = 0 \quad (5.127)$$

for the source distribution on panel i . Since only one strength and one parameter are involved in this case, the strength-parameter transformation is simply the identity ($[T_\sigma] = [I]$), and

$$\{\sigma\} = \{S\} \quad (5.128)$$

5.5 Transformations for Higher-Order Distributions

The strength-parameter transformations are more complicated for the higher-order distributions than those for the lower-order distributions. As such, various methods for calculating these transformations were attempted.

5.5.1 Quadratic Doublet Distribution

For a quadratic doublet distribution, six distribution parameters are needed (see Eq. (5.1)). In addition, continuity of doublet strength between panels should be maintained to avoid panel-edge vortices [50]. As part of this research effort, three different methods for determining the strength-parameter transformation were attempted.

Midpoint Addition Method

It was initially recognized that having doublet strengths defined at six unique points would provide a well-defined transformation between the strength and parameter spaces, since the parameter space for a quadratic distribution is six-dimensional. To provide the additional points (each panel already had the doublet strengths defined at each of the three vertices), the doublet strength at each edge midpoint was added as an unknown. The doublet strengths at the panel vertices and edge midpoints would then completely define the quadratic distribution.

The main advantage of this method was that it enforced doublet continuity across panel edges exactly. Since three points on a line uniquely determine a quadratic, and panels sharing an edge share the doublet strengths at the edge endpoints and midpoint, the distribution of doublet strength for each panel is exactly the same at each edge.

The primary disadvantage to this method, and the main reason for its failure, was the extreme computational cost it introduced. For an unstructured mesh with triangular panels, there tend to be approximately three to four times as many edges as there are vertices. Since this method introduced a new unknown on each panel edge, the size of the system of equations quintupled. For $O(N^2)$ operations, such as calculating influence

coefficients and solving the linear system of equations (using an efficient solver), this meant as much as a 25-fold increase in computation time. At best, the higher-order panels would be expected to increase the order of convergence by a factor of two, and so this increase in computation time was deemed unacceptable.

In addition, this method was very complicated to implement, requiring a significant amount of mesh manipulation and extra storage compared to using lower-order panels. The difference in code required between the two was significant, making code maintenance difficult.

Six-Point-Fit Method

The next attempted method also involved fitting a quadratic exactly to six points with defined doublet strengths. However, rather than adding unknowns at the edge midpoints, the additional unknowns were taken from the extra vertices of the three abutting panels, as shown in Fig. 5.1. Unlike the midpoint addition method, this method allowed for quadratic distributions over each panel without adding unknowns. Additionally, this method required only one modified subroutine in the pilot code compared to the lower-order distributions and very little additional memory.

Using the six-point-fit method, a quadratic doublet distribution over a panel is defined by

$$[S_\mu] \begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \\ \mu_{\eta\eta} \end{Bmatrix} = \begin{Bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \end{Bmatrix} \quad (5.129)$$

where $[S_\mu]$ is the matrix introduced in the previous section. In this case,

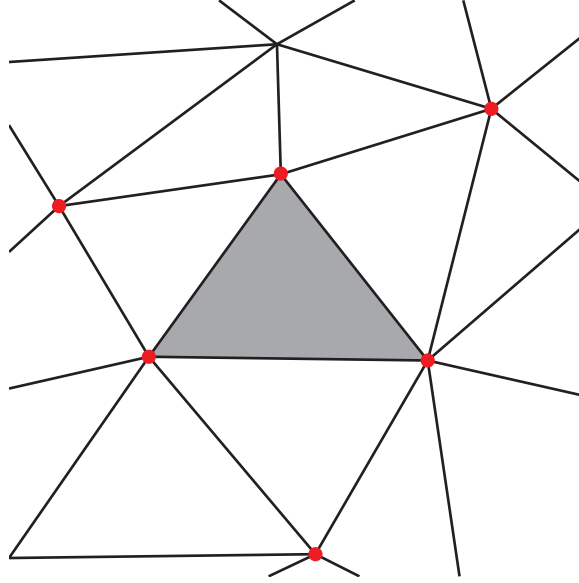


Fig. 5.1: Points used to define the quadratic doublet distribution over a panel in the six-point-fit method. The six mesh vertices marked in red are the points used for the gray panel.

$$[S_\mu] = \begin{bmatrix} 1 & \xi_1 & \eta_1 & \frac{1}{2}\xi_1^2 & \xi_1\eta_1 & \frac{1}{2}\eta_1^2 \\ 1 & \xi_2 & \eta_2 & \frac{1}{2}\xi_2^2 & \xi_2\eta_2 & \frac{1}{2}\eta_2^2 \\ 1 & \xi_3 & \eta_3 & \frac{1}{2}\xi_3^2 & \xi_3\eta_3 & \frac{1}{2}\eta_3^2 \\ 1 & \xi_4 & \eta_4 & \frac{1}{2}\xi_4^2 & \xi_4\eta_4 & \frac{1}{2}\eta_4^2 \\ 1 & \xi_5 & \eta_5 & \frac{1}{2}\xi_5^2 & \xi_5\eta_5 & \frac{1}{2}\eta_5^2 \\ 1 & \xi_6 & \eta_6 & \frac{1}{2}\xi_6^2 & \xi_6\eta_6 & \frac{1}{2}\eta_6^2 \end{bmatrix} \quad (5.130)$$

where (ξ_i, η_i) is the location in local-scaled coordinates of the point at which μ_i is defined. Points 4, 5, and 6 are the vertices on the neighboring panels opposite edges 1, 2, and 3, respectively. In this case, the transformation is again simply

$$[T_\mu] = [S_\mu]^{-1} \quad (5.131)$$

One disadvantage of the six-point-fit method was that it did not enforce continuity of doublet strength between panels. With the six-point-fit method, two abutting panels would share four points in defining their doublet distributions, but the other two points would be

unique for each. The exact effect of this was never explored due to the discovery of a more serious issue.

Using points from neighboring panels to define the doublet distribution meant that a certain level of continuity in the gradient of doublet strength was being created between panels. This amounted to greater continuity of surface velocities and pressures. In cases where the true solution is smooth and continuous, this is an advantage. However, in places where the surface velocity should be discontinuous, such as along sharp breaks in the geometry or along characteristic lines in supersonic flow, this heightened continuity was inappropriate. The velocities and pressures would not be exactly continuous between panels with this method, but there would be greater continuity than would be obtained with lower-order distributions.

A means was then developed to remove this pseudo-continuity in gradient at sharp breaks in the geometry while still maintaining continuity in doublet strength. First, a search was made to locate all edges on the mesh where the doublet gradient should be discontinuous. Whether an edge was labeled “discontinuous” was decided based on the dot product of the normal vectors to the two panels sharing the edge. If this dot product was below a certain user-defined threshold, the edge was marked as discontinuous. The threshold for discontinuous edges can be calculated as the cosine of an angle, where the angle represents the maximum flow-turning angle between panels for continuous edges. This angle is referred to as the *maximum continuity angle*.

At a panel level, if an edge was marked as discontinuous, then the additional vertex from the abutting panel for that edge was not used to determine the quadratic distribution (see Fig. 5.2). However, this made the transformation from strength space (now five-dimensional) to parameter space (still six-dimensional) non-unique. To make the transformation unique, an additional condition was needed. The chosen condition was that the doublet strength on the discontinuous edge should be exactly linear. This would exactly enforce continuity of doublet strength across the edge, since only one linear distribution will satisfy the required doublet strengths at both edge endpoints.

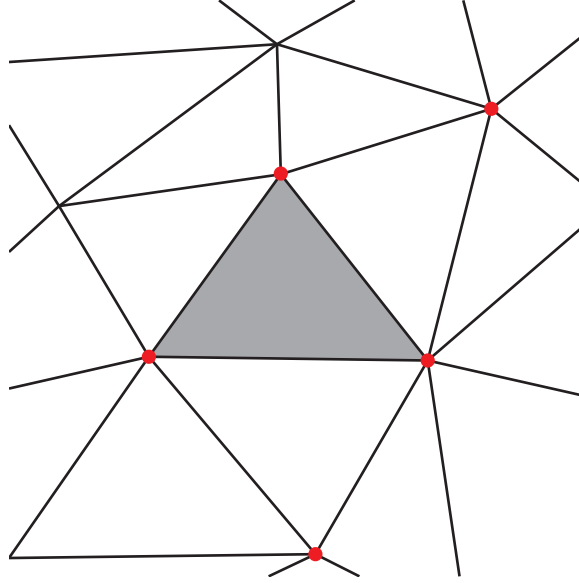


Fig. 5.2: Points used to defined the quadratic doublet distribution over a panel with one discontinuous edge in the six-point-fit method. The points used are highlighted in red. The left edge is discontinuous.

Through some manipulation, this condition may be expressed mathematically. Without loss of generality, assume the edge beginning at vertex 1 and ending at vertex 2 is discontinuous. For this edge, a position along its length may be described in local-scaled coordinates as a function of the displacement s by

$$\begin{bmatrix} \xi(s) \\ \eta(s) \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \eta_1 \end{bmatrix} + s \begin{bmatrix} \frac{\xi_2 - \xi_1}{l_{12}} \\ \frac{\eta_2 - \eta_1}{l_{12}} \end{bmatrix} \quad (5.132)$$

where

$$l_{12} = \sqrt{(\xi_2 - \xi_1)^2 + (\eta_2 - \eta_1)^2} \quad (5.133)$$

From this, the doublet distribution on the edge is given by

$$\mu(s) = \mu_0 + \mu_\xi \xi(s) + \mu_\eta \eta(s) + \mu_{\xi\xi} \xi(s)^2/2 + \mu_{\xi\eta} \xi(s)\eta(s) + \mu_{\eta\eta} \eta(s)^2/2 \quad (5.134)$$

If the doublet strength is linear along the edge, then it must be that

$$\frac{d^2\mu}{ds^2} = 0 \quad (5.135)$$

Taking the derivative of Eq. (5.134) with respect to s yields

$$\frac{d\mu}{ds} = \mu_\xi \frac{d\xi}{ds} + \mu_\eta \frac{d\eta}{ds} + \mu_{\xi\xi} \xi(s) \frac{d\xi}{ds} + \mu_{\xi\eta} \left(\eta(s) \frac{d\xi}{ds} + \xi(s) \frac{d\eta}{ds} \right) + \mu_{\eta\eta} \eta(s) \frac{d\eta}{ds} \quad (5.136)$$

From Eq. (5.132)

$$\frac{d\xi}{ds} = \frac{\xi_2 - \xi_1}{l_{12}} = \hat{t}_\xi \quad (5.137)$$

$$\frac{d\eta}{ds} = \frac{\eta_2 - \eta_1}{l_{12}} = \hat{t}_\eta \quad (5.138)$$

where $\hat{\mathbf{t}} = [\hat{t}_\xi, \hat{t}_\eta]^T$ is the tangent vector to the edge, which is constant. Thus

$$\frac{d\mu}{ds} = \mu_\xi \hat{t}_\xi + \mu_\eta \hat{t}_\eta + \mu_{\xi\xi} \xi(s) \hat{t}_\xi + \mu_{\xi\eta} (\eta(s) \hat{t}_\xi + \xi(s) \hat{t}_\eta) + \mu_{\eta\eta} \eta(s) \hat{t}_\eta \quad (5.139)$$

Taking the derivative again and applying Eq. (5.135) results in

$$0 = \mu_{\xi\xi} \hat{t}_\xi^2 + 2\mu_{\xi\eta} \hat{t}_\xi \hat{t}_\eta + \mu_{\eta\eta} \hat{t}_\eta^2 \quad (5.140)$$

as being a sufficient condition for $\mu(s)$ being linear on the edge.

Equation (5.140) may be manipulated so that it can be used in the transformation from strength space to parameter space. Without loss of generality, consider now the case where edge 3 is discontinuous. In this case, μ_6 cannot be used and the doublet strength on that edge must be linear. Since μ_6 is not used, that row is removed from $[S_\mu]$. From before, the condition that μ be linear along edge 3 is given by

$$0 = \mu_{\xi\xi}\hat{t}_{3\xi}^2 + 2\mu_{\xi\eta}\hat{t}_{3\xi}\hat{t}_{3\eta} + \mu_{\eta\eta}\hat{t}_{3\eta}^2 \quad (5.141)$$

In order to use this to determine the doublet distribution parameters from the doublet strengths, one of the unknown parameters must be solved for. This results in the three relations

$$\mu_{\xi\xi} = -2\mu_{\xi\eta}\frac{\hat{t}_{3\eta}}{\hat{t}_{3\xi}} - \mu_{\eta\eta}\frac{\hat{t}_{3\eta}^2}{\hat{t}_{3\xi}^2} \quad (5.142)$$

$$\mu_{\xi\eta} = -\frac{1}{2}\frac{\hat{t}_{3\xi}}{\hat{t}_{3\eta}} - \frac{1}{2}\frac{\hat{t}_{3\eta}}{\hat{t}_{3\xi}} \quad (5.143)$$

$$\mu_{\eta\eta} = -2\mu_{\xi\eta}\frac{\hat{t}_{3\xi}}{\hat{t}_{3\eta}} - \mu_{\xi\xi}\frac{\hat{t}_{3\xi}^2}{\hat{t}_{3\eta}^2} \quad (5.144)$$

The specific relation to use is chosen in order to avoid division by a small quantity. If $|\hat{t}_{3\xi}| < |\hat{t}_{3\eta}|$, then Eq. (5.144) should be used. Otherwise, Eq. (5.142) should be used. Since Eq. (5.143) requires dividing by both $\hat{t}_{3\xi}$ and $\hat{t}_{3\eta}$, it should be avoided.

These relations may be used to increase the dimension of $\{\mu\}$ in the case where it has an incomplete set of parameters, as in

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{\hat{t}_{3\xi}^2}{\hat{t}_{3\eta}^2} & -2\frac{\hat{t}_{3\xi}}{\hat{t}_{3\eta}} \end{bmatrix} \begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \end{Bmatrix} = \begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \\ \mu_{\eta\eta} \end{Bmatrix} \quad (5.145)$$

The matrix in this equation is referred to as $[A_{\eta\eta_3}]$ as it adds in $\mu_{\eta\eta}$ using the linearity condition on edge 3. Combining this with $[S_\mu]$ results in a relationship between the strength space and the incomplete parameter space, given by

$$[S_\mu][A_{\eta\eta_3}] \begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \end{Bmatrix} = \begin{Bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \end{Bmatrix} \quad (5.146)$$

Since $[S_\mu][A_{\eta\eta_3}]$ is square, it may be inverted. Premultiplying by $[A_{\eta\eta_3}]$ then completes the transformation from the strength space to the complete parameter space, given by

$$\{\mu\} = [A_{\eta\eta_3}]([S_\mu][A_{\eta\eta_3}])^{-1}\{M\} \quad (5.147)$$

This generalizes to other edges being discontinuous and other parameters being eliminated.

At this point it is good to consider what is happening in Eq. (5.147). What is needed are the six distribution parameters ($\{\mu\}$) that will dictate the influence of the panel under consideration. Premultiplying $\{M\}$ by $([S_\mu][A_{\eta\eta_3}])^{-1}$ will give the incomplete set of parameters that produce the correct doublet strengths at each of the points represented in $\{M\}$, as well as satisfy the linearity condition on edge 3. Premultiplying this result by $[A_{\eta\eta_3}]$ then adds in the missing distribution parameter using the linearity condition.

If two edges are discontinuous, the strength space is four-dimensional. Two $[A]$ matrices are used in this case, each one adding a parameter based on the condition of linearity on each of the discontinuous edges.

Unfortunately, there is a fundamental issue with the six-point-fit method. This issue comes from the linearity condition along the discontinuous edges. Observing the form of Eq. (5.140), it can be seen that the linearity condition is enforced, not just on the discontinuous edge, but everywhere on the panel surface. There is no spatial dependence in Eq. (5.140), and so the doublet strength is forced to be linear everywhere in the direction parallel to the discontinuous edge.

This is not always an issue. Whether this becomes an issue is dependent upon the mesh used. Consider the mesh shown in Fig. 5.3. The top edge of this mesh is a sharp

leading edge with the panels on the opposite face of the body not shown. Because the edge is sharp, the upper edge of the highlighted panel will be marked as discontinuous in the six-point-fit method. Thus, the doublet strength distribution is to be determined from the five red points and the condition that the doublet strength on any line parallel to the upper edge be linear.

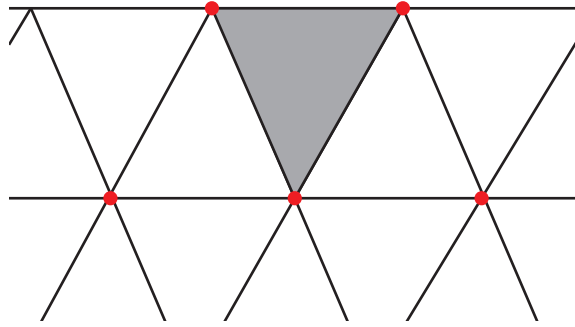


Fig. 5.3: Mesh paneling for which the six-point-fit method fails. The gray panel is the one under consideration, and the five red vertices are those used to determine the panel's doublet strength distribution. The top edge in this figure is considered discontinuous.

The issue is that the lower three points are colinear and parallel to the discontinuous edge. Because of the linearity condition imposed parallel to the discontinuous edge, the doublet strengths at these three points must follow an exact linear distribution. However, this cannot be satisfied for arbitrary doublet strengths at the three points. The transformation only has control over the distribution parameters based on the doublet strengths; it cannot specify the doublet strengths themselves. Hence, the strength to parameter transformation becomes singular for this mesh.

This could be overcome by remeshing, but this is undesirable. A robust method should produce a valid result regardless of the structure of the mesh it is given (within reasonable constraints). In addition, this is not an issue of mesh sensitivity where a bad result is obtained due to the mesh. Rather, if the strength to parameter transformation is singular, then a result cannot be computed at all, as the AIC matrix in Eq. (2.22) will contain

NaNs and cannot be inverted. The case outlined above is likely to occur on any number of different configurations, and so the six-point-fit cannot be used.

Midpoint-Interpolation Method

Before introducing the final method investigated, the strengths of the previous methods should be highlighted. The midpoint addition method allowed for exact doublet strength continuity along panel edges. The six-point-fit method allowed for higher-order distributions without adding extra unknowns. Here, these advantages are combined in what is called the midpoint-interpolation method.

As seen before, if the doublet strength is defined at each panel midpoint, then an exact, analytic distribution of doublet strength across each panel can be defined that automatically satisfies continuity with neighboring panels. However, rather than adding the midpoint strengths as additional unknowns, the midpoint strengths may be determined by interpolating the doublet strengths at neighboring mesh vertices.

As with the six-point-fit method, using neighboring vertices to help determine the doublet strength distribution introduces some potentially-undesirable continuity in the solution. As such, the idea of discontinuous edges is considered here as well.

Consider first the case of a panel with all continuous edges. This panel has vertices 1, 2, and 3. Midpoint 1 lies on the edge between vertices 1 and 2. The vertex opposite this edge on the panel abutting the panel under consideration will be called vertex 4. These four vertices will be used in determining the doublet strength at midpoint 1, as shown in Fig. 5.4a. If these four vertices are used, then the interpolated strength at the midpoint will be the same for both panels sharing that edge, and the quadratic doublet strength will be exactly continuous between the two panels (a significant improvement over the six-point-fit method).

The question is now how to determine the midpoint strength as a function of the four vertex strengths under consideration. The most straightforward method for doing this is to fit a quadratic to the doublet strengths at these four vertices. Writing out the doublet

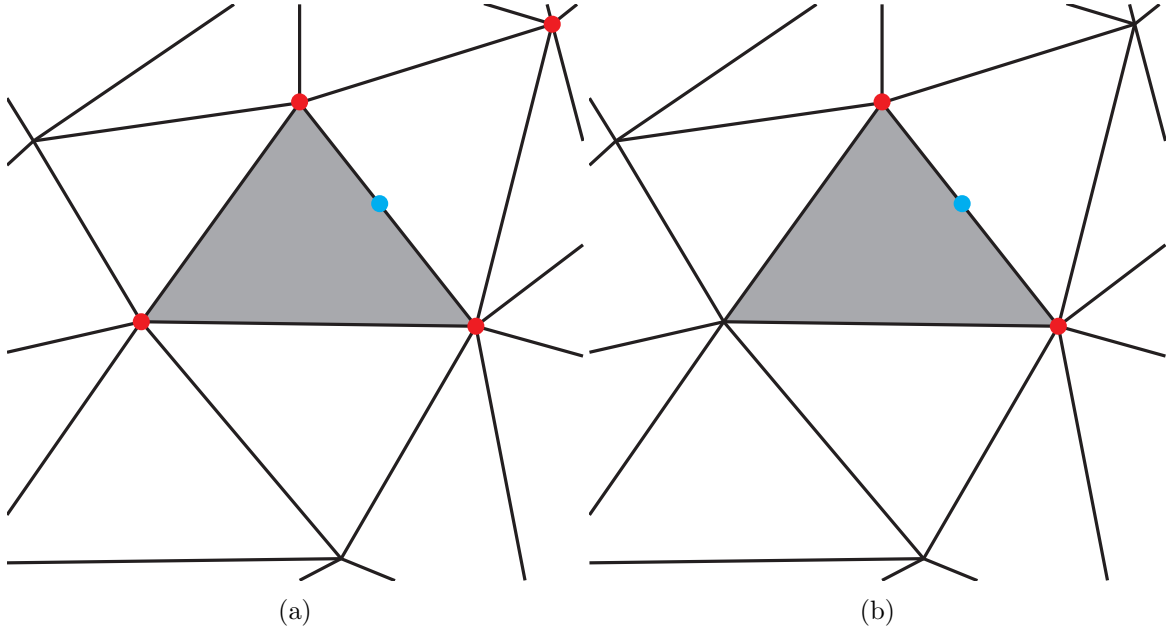


Fig. 5.4: Points used (red) to determine midpoint strengths (blue) for a) continuous and b) discontinuous edges.

strength predicted at each relevant vertex by a quadratic distribution yields the matrix equation

$$\begin{bmatrix} 1 & \xi_1 & \eta_1 & \frac{1}{2}\xi_1^2 & \xi_1\eta_1 & \frac{1}{2}\eta_1^2 \\ 1 & \xi_2 & \eta_2 & \frac{1}{2}\xi_2^2 & \xi_2\eta_2 & \frac{1}{2}\eta_2^2 \\ 1 & \xi_3 & \eta_3 & \frac{1}{2}\xi_3^2 & \xi_3\eta_3 & \frac{1}{2}\eta_3^2 \\ 1 & \xi_4 & \eta_4 & \frac{1}{2}\xi_4^2 & \xi_4\eta_4 & \frac{1}{2}\eta_4^2 \end{bmatrix} \begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \\ \mu_{\eta\eta} \end{Bmatrix} = \begin{Bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{Bmatrix} \quad (5.148)$$

where (ξ_i, η_i) is the location of vertex i in local-scaled coordinates, and μ_i is the doublet strength at vertex i . This equation is underdetermined, but a minimum-norm solution may be found using least-squares¹. The minimum norm solution to this equation is given by [126]

¹The minimum norm solution for an underdetermined system is the solution that both satisfies the system of equations and minimizes the norm of the solution vector. For more on minimum-norm solutions using least-squares, see section 3.15 in [126].

$$\begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \\ \mu_{\eta\eta} \end{Bmatrix} = [\Sigma]^T ([\Sigma][\Sigma]^T)^{-1} \begin{Bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{Bmatrix} \quad (5.149)$$

where

$$[\Sigma] = \begin{bmatrix} 1 & \xi_1 & \eta_1 & \frac{1}{2}\xi_1^2 & \xi_1\eta_1 & \frac{1}{2}\eta_1^2 \\ 1 & \xi_2 & \eta_2 & \frac{1}{2}\xi_2^2 & \xi_2\eta_2 & \frac{1}{2}\eta_2^2 \\ 1 & \xi_3 & \eta_3 & \frac{1}{2}\xi_3^2 & \xi_3\eta_3 & \frac{1}{2}\eta_3^2 \\ 1 & \xi_4 & \eta_4 & \frac{1}{2}\xi_4^2 & \xi_4\eta_4 & \frac{1}{2}\eta_4^2 \end{bmatrix} \quad (5.150)$$

The doublet strength at midpoint 1 may then be found by combining the coordinates of the midpoint, (ξ_{m_1}, η_{m_1}) , with the coefficients just determined, as in

$$\mu_{m_1} = \begin{bmatrix} 1 & \xi_{m_1} & \eta_{m_1} & \frac{1}{2}\xi_{m_1}^2 & \xi_{m_1}\eta_{m_1} & \frac{1}{2}\eta_{m_1}^2 \end{bmatrix} \begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \\ \mu_{\eta\eta} \end{Bmatrix} \quad (5.151)$$

Let

$$\mathbf{d}_{m_i}^T = \begin{bmatrix} 1 & \xi_{m_i} & \eta_{m_i} & \frac{1}{2}\xi_{m_i}^2 & \xi_{m_i}\eta_{m_i} & \frac{1}{2}\eta_{m_i}^2 \end{bmatrix} \quad (5.152)$$

The midpoint strength may then be written in terms of the vertex strengths by using Eq. (5.149), giving

$$\mu_{m_1} = \mathbf{d}_{m_i}^T [\Sigma]^T ([\Sigma][\Sigma]^T)^{-1} \begin{Bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{Bmatrix} \quad (5.153)$$

Thus, the strengths at each midpoint may then be found from the doublet strengths at the four neighboring vertices. For each midpoint, the vertices used to determine its doublet strength are the same for both panels sharing that midpoint. Thus, the midpoint strength is the same for abutting panels, and the doublet strength is exactly continuous between panels.

As with a linear distribution, writing the doublet distribution equation for each vertex and midpoint results in a matrix equation of the form

$$\begin{bmatrix} 1 & \xi_1 & \eta_1 & \frac{1}{2}\xi_1^2 & \xi_1\eta_1 & \frac{1}{2}\eta_1^2 \\ 1 & \xi_2 & \eta_2 & \frac{1}{2}\xi_2^2 & \xi_2\eta_2 & \frac{1}{2}\eta_2^2 \\ 1 & \xi_3 & \eta_3 & \frac{1}{2}\xi_3^2 & \xi_3\eta_3 & \frac{1}{2}\eta_3^2 \\ 1 & \xi_{m_1} & \eta_{m_1} & \frac{1}{2}\xi_{m_1}^2 & \xi_{m_1}\eta_{m_1} & \frac{1}{2}\eta_{m_1}^2 \\ 1 & \xi_{m_2} & \eta_{m_2} & \frac{1}{2}\xi_{m_2}^2 & \xi_{m_2}\eta_{m_2} & \frac{1}{2}\eta_{m_2}^2 \\ 1 & \xi_{m_3} & \eta_{m_3} & \frac{1}{2}\xi_{m_3}^2 & \xi_{m_3}\eta_{m_3} & \frac{1}{2}\eta_{m_3}^2 \end{bmatrix} \begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \\ \mu_{\eta\eta} \end{Bmatrix} = \begin{Bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_{m_1} \\ \mu_{m_2} \\ \mu_{m_3} \end{Bmatrix} \quad (5.154)$$

The matrix in this equation is called $[S_\mu]$, following the convention established previously. Inverting this equation will yield the doublet parameters in terms of the doublet strengths, as in

$$\begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \\ \mu_{\eta\eta} \end{Bmatrix} = [S_\mu]^{-1} \begin{Bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_{m_1} \\ \mu_{m_2} \\ \mu_{m_3} \end{Bmatrix} \quad (5.155)$$

Using the equations just developed to express the midpoint strengths in terms of the neighboring vertex strengths, this may be written purely in terms of vertex doublet strengths

$$\begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \\ \mu_{\eta\eta} \end{Bmatrix} = [S_\mu]^{-1} \begin{bmatrix} 1 & 0 & & 0 & 0 & 0 \\ 0 & 1 & & 0 & 0 & 0 \\ 0 & 0 & & 1 & 0 & 0 \\ -- & -- & \mathbf{d}_{m_1}^T [\Sigma]^t ([\Sigma][\Sigma]^T)^{-1} [I'_4] & -- & -- & -- \\ -- & -- & \mathbf{d}_{m_2}^T [\Sigma]^t ([\Sigma][\Sigma]^T)^{-1} [I'_5] & -- & -- & -- \\ -- & -- & \mathbf{d}_{m_3}^T [\Sigma]^t ([\Sigma][\Sigma]^T)^{-1} [I'_6] & -- & -- & -- \end{bmatrix} \begin{Bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \end{Bmatrix} \quad (5.156)$$

where $[I'_i]$ is a 4×6 matrix with a 3×3 identity in the upper-left corner and the i -th element of row 4 equal to 1. For example,

$$[I'_5] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.157)$$

The matrix $[I'_i]$ serves to ensure the correct vertices influence the midpoint strength interpolation. Each $[\Sigma]$ matrix in the above equation is dependent upon which vertices are used to interpolate each midpoint. The first three rows of $[\Sigma]$ are always identical, as vertices 1, 2, and 3 are always used to interpolate the midpoint strength. Recognizing this allows for some computational savings.

For a discontinuous edge, the midpoint strength is simply assumed to be the average of the two neighboring endpoint strengths (see Fig. 5.4b). This exactly enforces continuity and linearity on that edge, the respective advantages of the previous two methods. Continuity will be satisfied because the midpoint strength is calculated using the same two vertices for both panels sharing the edge. The doublet distribution will be linear on that edge because

the midpoint is exactly halfway between the two end vertices and its strength is the average of theirs, making a quadratic distribution impossible.

If edge i is discontinuous, then row $i + 3$ of the matrix in Eq. (5.156) is all zero except for $\frac{1}{2}$ in the two columns corresponding to the two edge end vertices. For example, if edge 2 is discontinuous, the transformation looks like

$$\begin{Bmatrix} \mu_0 \\ \mu_\xi \\ \mu_\eta \\ \mu_{\xi\xi} \\ \mu_{\xi\eta} \\ \mu_{\eta\eta} \end{Bmatrix} = [S_\mu]^{-1} \begin{bmatrix} 1 & 0 & & 0 & 0 & 0 \\ 0 & 1 & & 0 & 0 & 0 \\ 0 & 0 & & 1 & 0 & 0 \\ -- & -- & \mathbf{d}_{m_1}^T [\Sigma]^t ([\Sigma][\Sigma]^T)^{-1} [I'_4] & -- & -- & -- \\ 0 & \frac{1}{2} & & \frac{1}{2} & 0 & 0 \\ -- & -- & \mathbf{d}_{m_3}^T [\Sigma]^t ([\Sigma][\Sigma]^T)^{-1} [I'_6] & -- & -- & -- \end{bmatrix} \begin{Bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \end{Bmatrix} \quad (5.158)$$

If all edges of a panel are discontinuous, then that panel should simply be modeled with a linear doublet distribution. An advantage to the midpoint-interpolation method is that the implementation of quadratic and linear doublet distributions differs only in one subroutine. Thus, mixing of panel orders within the same mesh would be very easy.

5.5.2 Linear Source Distribution

Two approaches to higher-order sources were attempted in this research effort. Unlike with doublet distributions, the source strength is not required to be continuous between panels. As such, no effort was made here to enforce continuity.

Vertex-Based

Initially, it was recognized that an exact distribution of linear source strength could be achieved everywhere on the geometry if the source strengths were simply defined at the panel vertices, such as with a linear doublet distribution. However, this produced poor results at points on the mesh with large angles between panels. As described in Chapter 2, the source strengths are determined based on the angle between the configuration surface

and the freestream velocity. At each vertex, this required averaging the normal vector between the panels neighboring that vertex. This also meant that panels on either side of a sharp edge shared a continuous source distribution. Since the source distribution represents a jump in velocity directly, the velocities were forced to be continuous across such locations in the mesh. This was undesirable, as the velocity should not be continuous across a sharp edge, particularly in supersonic flow.

Allowing for discontinuities (such as with the doublet distributions) would have required creating duplicate mesh vertices along discontinuous edges. These duplicate vertices would have had differing source strengths but would still need to have identical doublet strengths so as to satisfy doublet continuity. It was determined that the computational overhead for this method was too great, and the method was abandoned.

Neighboring Least-Squares

Based on the difficulties with vertex-based source strengths just described, it was decided that the linear source distributions should be determined by interpolating the source strengths from the centroids of neighboring panels, rather than having the source strengths be vertex-based. This simplifies implementation, as the unknown source strengths are still the source strengths at the centroid of each panel, the same as with constant-source panels. This method also easily accommodates discontinuous edges, as panels across discontinuous edges can simply be ignored in the interpolation. And, with the linear source distributions based on neighboring panel strengths, no extra unknowns need to be added when moving from constant to linear source panels.

When fitting a linear distribution from the source strengths at the centroid of neighboring panels, there are four source strengths available (assuming no discontinuous edges): the source strength at the centroid of the panel under consideration and those for its three neighbors. Since a linear distribution requires only three distribution parameters, the strength-parameter transformation is thus over-determined. Two methods were examined as part of this research for overcoming this.

The first and simplest method considered involves using least-squares approximation to determine the distribution parameters that will provide the best fit to the four source strengths. Writing Eq. (5.3) for each of the relevant source strengths yields the over-determined matrix equation

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & \xi_{n_1} & \eta_{n_1} \\ 1 & \xi_{n_2} & \eta_{n_2} \\ 1 & \xi_{n_3} & \eta_{n_3} \end{bmatrix} \begin{Bmatrix} \sigma_0 \\ \sigma_\xi \\ \sigma_\eta \end{Bmatrix} = \begin{Bmatrix} \sigma_i \\ \sigma_{n_1} \\ \sigma_{n_2} \\ \sigma_{n_3} \end{Bmatrix} \quad (5.159)$$

where (ξ_{n_i}, η_{n_i}) is the location in local-scaled coordinates of the centroid of neighboring panel i , σ_i is the source strength at the centroid of the panel under consideration, and σ_{n_i} is the source strength at the centroid of the i -th neighbor to the panel under consideration. The first row of this matrix has mostly zeros since $(\xi, \eta) = (0, 0)$ is the centroid of the panel.

If the matrix $[S_\sigma]$ is defined as

$$[S_\sigma] = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \xi_{n_1} & \eta_{n_1} \\ 1 & \xi_{n_2} & \eta_{n_2} \\ 1 & \xi_{n_3} & \eta_{n_3} \end{bmatrix} \quad (5.160)$$

then the least-squares solution to the over-determined system of Eq. (5.159) is given by [126]

$$\begin{Bmatrix} \sigma_0 \\ \sigma_\xi \\ \sigma_\eta \end{Bmatrix} = ([S_\sigma]^T [S_\sigma])^{-1} [S_\sigma]^T \begin{Bmatrix} \sigma_i \\ \sigma_{n_1} \\ \sigma_{n_2} \\ \sigma_{n_3} \end{Bmatrix} \quad (5.161)$$

This method was originally used in MachLine for higher-order source distributions. However, it was found that, when higher-order distributions were used, the Morino formulation resulted in a much lower rate of grid convergence than the source-free formulation.

This method for implementing higher-order source fits was suspected as being the reason for the poor grid convergence.

The underlying cause for this is that Eq. (5.161) will not necessarily result in $\sigma_0 = \sigma_i$ for a given panel. That is, the source strength at the panel centroid predicted as part of the distribution parameters is not necessarily the same as the source strength already defined at the panel centroid. In fact, for certain panel configurations, it was possible with this method for $\sigma(\xi, \eta)$ to not be equal to σ_i anywhere on the panel. This was clearly in error.

Thus, a method was sought that would explicitly force $\sigma_0 = \sigma_i$. This can be achieved by setting σ_0 directly, and then determining σ_ξ and σ_η using least-squares based on the neighboring panel source strengths. Writing Eq. (5.3) for each of the panel neighbors, with $\sigma_0 = \sigma_i$, results in

$$\sigma_0 + \begin{bmatrix} \xi_{n_1} & \eta_{n_1} \\ \xi_{n_2} & \eta_{n_2} \\ \xi_{n_3} & \eta_{n_3} \end{bmatrix} \begin{Bmatrix} \sigma_\xi \\ \sigma_\eta \end{Bmatrix} = \begin{Bmatrix} \sigma_{n_1} \\ \sigma_{n_2} \\ \sigma_{n_3} \end{Bmatrix} \quad (5.162)$$

This system for calculating σ_ξ and σ_η is still over-determined. Thus, a least-squares solution may be used, resulting in

$$\begin{Bmatrix} \sigma_\xi \\ \sigma_\eta \end{Bmatrix} = ([\Sigma]^T [\Sigma])^{-1} [\Sigma]^T \begin{Bmatrix} \sigma_{n_1} - \sigma_i \\ \sigma_{n_2} - \sigma_i \\ \sigma_{n_3} - \sigma_i \end{Bmatrix} \quad (5.163)$$

where

$$[\Sigma] = \begin{bmatrix} \xi_{n_1} & \eta_{n_1} \\ \xi_{n_2} & \eta_{n_2} \\ \xi_{n_3} & \eta_{n_3} \end{bmatrix} \quad (5.164)$$

Recall that

$$\{S\} = \begin{Bmatrix} \sigma_i \\ \sigma_{n_1} \\ \sigma_{n_2} \\ \sigma_{n_3} \end{Bmatrix} \quad (5.165)$$

Thus, Eq. (5.163) may be written

$$\begin{Bmatrix} \sigma_\xi \\ \sigma_\eta \end{Bmatrix} = ([\Sigma]^T [\Sigma])^{-1} [\Sigma]^T \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \{S\} \quad (5.166)$$

Adding in the fact $\sigma_0 = \sigma_i$ yields the strength-parameter transformation

$$\begin{Bmatrix} \sigma_0 \\ \sigma_\xi \\ \sigma_\eta \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ ([\Sigma]^T [\Sigma])^{-1} [\Sigma]^T \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \end{bmatrix} \{S\} \quad (5.167)$$

Unlike the previous strength-parameter transformation, this transformation ensures that the source strength predicted at the centroid of the panel by the source distribution exactly matches the source strength calculated for that panel. It is this method where $\sigma_0 = \sigma_i$ that has been implemented in MachLine. Doing so, the rates of grid convergence for the Morino formulation using higher-order singularity distributions were found to be on par with those resulting from the source-free formulation.

Consider now the possibility of discontinuous edges, as discussed with the higher-order doublet transformations. If one of the edges of the panel is discontinuous, then there are only three source strengths from which to calculate the distribution parameters. In this case, the strength-parameter transformation may be obtained simply by writing Eq. (5.3) for each relevant source strength and inverting the system matrix. For example, if the third edge is discontinuous, this results in

$$\begin{Bmatrix} \sigma_0 \\ \sigma_\xi \\ \sigma_\eta \end{Bmatrix} = \left(\begin{bmatrix} 1 & 0 & 0 \\ 1 & \xi_{n_1} & \eta_{n_1} \\ 1 & \xi_{n_2} & \eta_{n_2} \end{bmatrix} \right)^{-1} \begin{Bmatrix} \sigma_i \\ \sigma_{n_1} \\ \sigma_{n_2} \end{Bmatrix} \quad (5.168)$$

If two of the edges of the panel are discontinuous, then there are only two source strengths from which to calculate the three distribution parameters, and an auxiliary condition is needed. A natural choice would be to specify that the source strength is constant in the direction perpendicular to the offset between the two influencing panel centroids. This is natural because there is no information in that direction. If the second and third edges are discontinuous, this condition is expressed as

$$\eta_{n_1} \sigma_\xi - \xi_{n_1} \sigma_\eta = 0$$

or simply

$$\eta_{n_1} \sigma_\xi = \xi_{n_1} \sigma_\eta$$

Solving for either distribution parameter, this becomes

$$\sigma_\xi = \frac{\xi_{n_1}}{\eta_{n_1}} \sigma_\eta \quad (5.169)$$

$$\sigma_\eta = \frac{\eta_{n_1}}{\xi_{n_1}} \sigma_\xi \quad (5.170)$$

As with the six-point-fit method for quadratic doublet panels, one of the above relations is used to eliminate one of the unknown distribution parameters. Since division by small numbers should be avoided, Eq. (5.169) is used if $|\xi_{n_1}| < |\eta_{n_1}|$, and Eq. (5.170) is used otherwise.

The transformation from $\{S\}$ space to $\{\sigma\}$ space now involves increasing the dimension by one using the condition described above. The general transformation is given by

$$\begin{Bmatrix} \sigma_0 \\ \sigma_\xi \\ \sigma_\eta \end{Bmatrix} = [A] ([S_\sigma][A])^{-1} \begin{Bmatrix} \sigma_i \\ \sigma_{n_1} \\ \sigma_{n_2} \end{Bmatrix} \quad (5.171)$$

where $[A]$ is dependent upon which parameter is being eliminated. If $|\xi_{n_1}| < |\eta_{n_1}|$ (σ_ξ is being eliminated), then $[A]$ is given by

$$[A] = \begin{bmatrix} 1 & 0 \\ 0 & \frac{\xi_{n_1}}{\eta_{n_1}} \\ 0 & 1 \end{bmatrix} \quad (5.172)$$

Otherwise (σ_η is being eliminated), $[A]$ is given by

$$[A] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & \frac{\eta_{n_1}}{\xi_{n_1}} \end{bmatrix} \quad (5.173)$$

The transformation is similarly derived for any other combination of two discontinuous edges.

For a panel with all discontinuous edges, the source distribution is simply constant.

5.6 Wake Modeling with Higher-Order Distributions

Wake panels only have distributions of doublet strength and no sources. Using the methods described previously, and assuming the wake shedding angle is greater than the maximum continuity angle (a reasonable assumption), any edge which sheds a wake will have a linear distribution of doublet strength along its length. Because of this, the doublet strength on the wake is simply linear.

This is not required, however. Since the doublet distribution continues from the body to the wake, the pilot code could be altered such that wake panels are allowed to have a quadratic distribution of doublet strength. The doublet strength would still have to be constant in the streamwise direction but could be allowed to vary quadratically in the

spanwise direction. This would require an auxiliary condition for fixing the midpoint doublet strength on each wake shedding edge.

The difficulty with determining the doublet distribution on each wake panel is that the wake strength is the difference between the wake strengths on the upper and lower surfaces bordering the wake-shedding edge. Within MachLine, each vertex in the mesh stores the indices of the top and bottom mesh vertices that determine its strength. Then, when the influence of a wake panel is calculated, it is incorporated into the linear system of equations by adding the influence for the top vertices and subtracting the influence for the bottom vertices.

CHAPTER 6

SOLVING THE LINEAR SYSTEM OF EQUATIONS

Never explicitly invert a matrix numerically. - Todd Moon and Wynn
Stirling [126]

Once Eq. (2.22) has been assembled, it must be solved to determine the unknown doublet strengths. For subsonic flows (i.e. $M_\infty < 1$) the Prandtl-Glauert equation is elliptic. In this case, each panel exerts an influence everywhere in the computational domain. Because of this, the linear system of equations entirely filled-in. The efficient solution of the system of equations in an elliptic BEM has been investigated previously (e.g. see [95–104]). However, the efficient solution of the system of equations in a supersonic panel method has not yet been considered. In supersonic flow, the Prandtl-Glauert equation is hyperbolic, and, as discussed previously, the influence of a given panel is limited to a downstream region known as the domain of influence. Because of this limited influence, the AIC matrix contains many elements that are identically zero. It is important to consider this fact when attempting to solve the linear system of equations efficiently. In this chapter, the efficient solution of the linear system of equations arising in the supersonic case is considered.

In this chapter, unless otherwise stated, results were obtained using the Morino boundary condition formulation and lower-order singularity distributions. The AIC matrix is identical for both boundary condition formulations, and so there should be no difference in solver performance between the two formulations. The effect of higher-order singularity distributions is more complex and will be discussed where necessary.

6.1 Structure of the Supersonic AIC Matrix

The structure of the AIC matrix is such that the rows correspond to which control point is being influenced, and the columns correspond to which vertex is exerting an influence. This is show diagrammatically in Fig. 6.1.

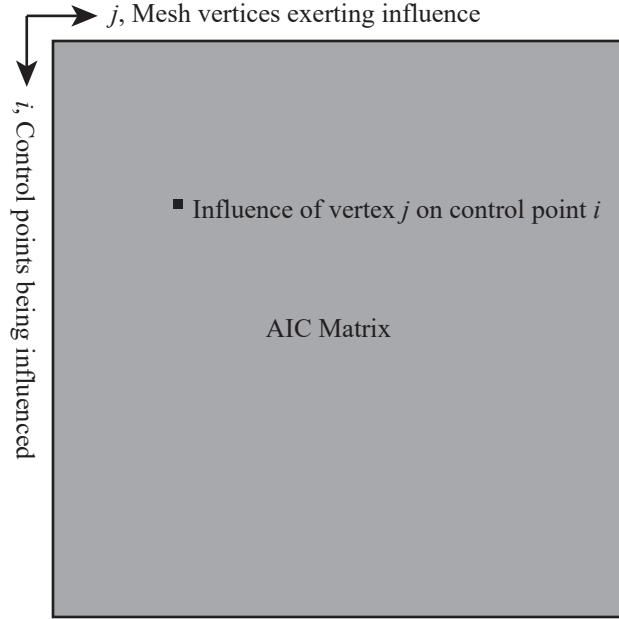


Fig. 6.1: Diagram of the general structure of an AIC matrix.

The difference between subsonic and supersonic AIC matrices is apparent in Fig. 6.2. The AIC matrix for subsonic flow (represented in Fig. 6.2a) has all non-zero elements. On the other hand, the supersonic AIC matrix for a straight wing (Fig. 6.2b) has mostly zero elements.

Initially, it was hypothesized that if the non-zero elements of the AIC matrix could be arranged into some convenient structure, then the matrix equation could be solved more efficiently. The ordering of the AIC matrix represented in Fig. 6.2b came from the order of the vertices in the mesh file; the unknowns are defined at each vertex, and each control

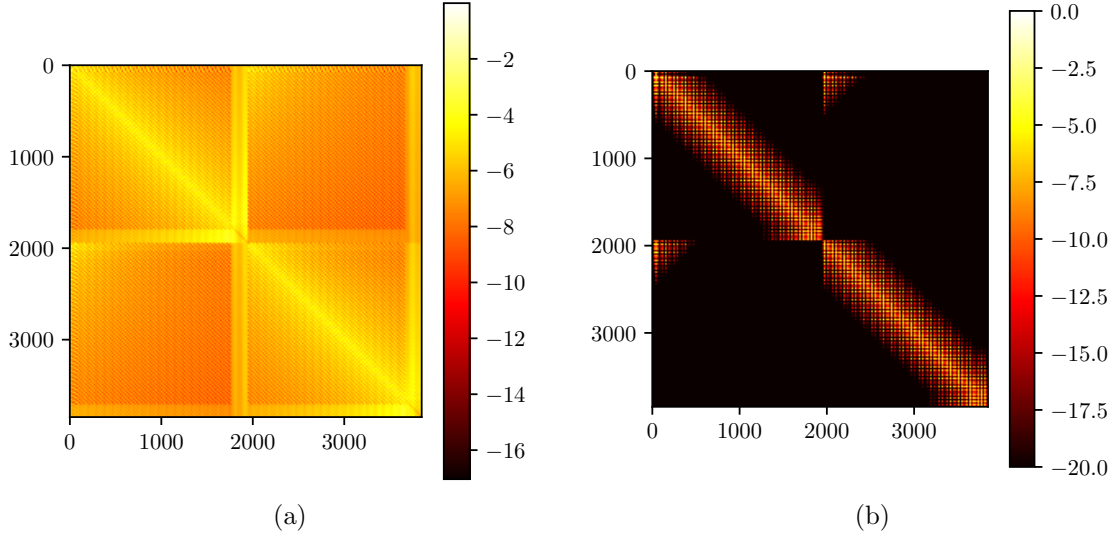


Fig. 6.2: Heatmaps showing the magnitudes of the AIC matrix coefficients for a) a standard subsonic case and b) a high-aspect-ratio, 5° half-angle, double-wedge wing in supersonic flow. The scale is logarithmic. Black regions signify AIC elements which are identically zero.

point is associated with a vertex. Because of this, there is no useful pattern to the non-zero elements represented in Fig. 6.2a.

Since the panels in supersonic flow exert no upstream influence, if the mesh vertices are arranged in a certain order relative to the freestream direction, then most of the non-zero elements in the AIC matrix could be made to appear in a single region of the matrix. This is evident in Fig. 6.3, which shows the AIC matrix for a 10° half-angle cone in supersonic flow. In the cone mesh file, the vertices proceed from most-downstream to most-upstream. Recalling the structure diagrammed in Fig. 6.1, the first row of the AIC matrix (corresponding to the most-downstream control point) for the cone is relatively filled-in, since that control point is influenced by most of the other vertices in the mesh. Moving down the rows of the AIC matrix (corresponding to moving upstream through control points), more and more elements on the left end of the row become zero. The zero elements are clustered at the left end of the row because those represent the influence of vertices which are further downstream.

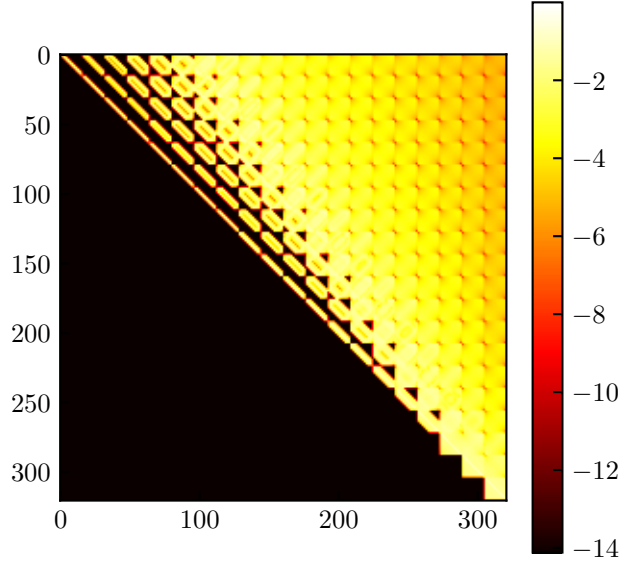


Fig. 6.3: Heatmap showing the magnitudes of the AIC matrix coefficients for a 10° half-angle cone in supersonic flow ($M_\infty = 1.5$). The scale is logarithmic. Black regions signify AIC elements which are identically zero.

Not all elements left of the diagonal in the AIC matrix of Fig. 6.3 are zero, meaning the AIC matrix is not upper-triangular. This is because a vertex may still have an influence on a point upstream of itself if that vertex belongs to a panel which does influence that upstream point. This comes from the way linear doublet distributions on panels are defined. The doublet strength at that vertex will influence the doublet distribution over the whole panel, meaning any point belonging to the panel's domain of influence may be influenced by that vertex, even if the point is outside the vertex's domain of influence. This influence is expanded even further upstream for higher-order panels using the midpoint-interpolation method, since vertices of neighboring panels are used to determine the quadratic doublet distribution.

Because of this limited upstream influence, the AIC matrix in Fig. 6.3 is *upper-pentagonal*, rather than upper-triangular. An upper-pentagonal matrix is defined by the following definition:

Definition 1 A square matrix, A , having elements A_{ij} is upper-pentagonal, iff

$$\exists B_l \geq 0 \quad \text{s.t.} \quad i > j + B_l \implies A_{ij} = 0 \quad \forall j \quad (6.1)$$

where the quantity B_l is called the lower bandwidth.

The lower bandwidth of an upper-pentagonal matrix is essentially a measure of the width of the region below the diagonal where non-zero elements appear. With this definition, it is immediately clear that the AIC matrix represented in Fig. 6.3 is upper-pentagonal.

However, a matrix simply being upper-pentagonal is not particularly useful. By Definition 1, an $N \times N$ matrix with all non-zero elements may be considered upper-pentagonal with bandwidth $B_l = N - 1$. As will be shown later, to efficiently solve the linear system of equations, it is best to have $B_l \ll N$. Since the ordering of the linear system is initially dependent upon the definition of the mesh, a method is sought that will permute the AIC matrix into upper-pentagonal form with small lower bandwidth.

The AIC matrix may be permuted into upper-pentagonal form based on a knowledge of the flow characteristics. As discussed previously, a given vertex (at which a doublet strength is defined) may exert an influence as far upstream as the most upstream vertex of any panel which it touches. Any points further upstream than this are guaranteed to not be influenced by the vertex under consideration. Consider now the mesh vertices being sorted based on the location of their most-upstream neighbor. If the first vertex in the mesh is the one whose most-upstream neighbor is furthest downstream, then that vertex would have the most limited influence out of all of the mesh vertices. This means the column of the AIC matrix corresponding to this vertex (the first column) would also have the fewest non-zero elements. The second vertex would then have the second-most limited influence, and the corresponding column of the AIC matrix (the second column) would have the second-fewest non-zero elements. And so on through all vertices in the mesh.

Permuting the AIC matrix as just described will ensure that the number of non-zero elements in each column increases from left to right within the matrix. However, this does nothing for setting the order of these non-zero elements within each column. To order the

rows of the AIC matrix, the control points being influenced should be considered. The most-downstream control point will most likely be the one influenced by the most mesh vertices, and so the rows of the AIC matrix should be arranged such that the furthest control point downstream is at the top of the AIC matrix and the furthest control point upstream is at the bottom of the AIC matrix. This will cluster the non-zero elements in each column of the AIC matrix at the top.

These two sorts, one based on most-upstream neighbors and the other based on control point locations, must be combined to determine the ordering of the AIC matrix. For simplicity, it is best to keep vertex i associated with control point i . This means in the AIC matrix that the vertex represented by the fifth column, say, is associated with the control point represented by the fifth row. Since the control points are placed very close to the mesh vertices in this implementation, the control points may simply be ordered based on the vertex locations.

The two sorts may be efficiently combined if the vertices are first sorted based purely on their location, and then sorted again based on most-upstream neighbors using a stable sorting algorithm. A stable sorting algorithm should be used so that the ordering from the first sort is preserved whenever the second sort encounters a tie. This will happen often since vertices will share upstream neighbors. The first sort will effectively sort the rows of the AIC matrix, and the second sort will finish arranging the columns of the AIC matrix.

This method is described in Algorithm 1. For efficiency, instead of rearranging the mesh vertices in memory, the sorting algorithm produces a permutation vector that gives the ordering of the rearranged vertices. The first step of the algorithm (lines 7-12) is to sort all vertices based on the negative of their downstream distance. If \mathbf{r}_i is the location of the i -th vertex, then the negative of the downstream distance, x_i , is given by

$$x_i = -\mathbf{U}_\infty \cdot \mathbf{r}_i \quad (6.2)$$

where \mathbf{U}_∞ is the freestream velocity vector.

Algorithm 1 Method for sorting mesh vertices such that the AIC matrix will be upper-pentagonal.

```

1: Input
2:    $N$    Number of mesh vertices
3:    $\mathbf{r}_i$  Vertex locations
4:    $\mathbf{U}_\infty$  Freestream velocity
5: Output
6:    $P$    Vector of vertex indices which will sort the AIC matrix (permutation vector)
7:    $i \leftarrow 1$ 
8:   while  $i \leq N$  do           ▷ Get the downstream distances of each vertex and store in  $x$ 
9:      $x_i \leftarrow -\langle \mathbf{U}_\infty, \mathbf{r}_i \rangle$ 
10:     $i \leftarrow i + 1$ 
11: end while
12:  $P_1 \leftarrow$  Indices which will sort  $x$ 
13:  $i \leftarrow 1$ 
14: while  $i \leq N$  do   ▷ Get the downstream distances of the most-downstream neighbors
15:    $j \leftarrow P_{1_i}$ 
16:    $k \leftarrow$  Index of most-downstream neighbor of vertex  $j$ 
17:    $x_i \leftarrow -\langle \mathbf{U}_\infty, \mathbf{r}_k \rangle$ 
18:    $i \leftarrow i + 1$ 
19: end while
20:  $P_2 \leftarrow$  Indices which will sort  $x$  using a stable sort
21:  $i \leftarrow 1$ 
22: while  $i \leq N$  do           ▷ Combine the two permutations
23:    $j \leftarrow P_{2_i}$ 
24:    $k \leftarrow P_{1_k}$ 
25:    $P_k \leftarrow i$ 
26:    $i \leftarrow i + 1$ 
27: end while

```

The second step of the algorithm (lines 13-20) is to then sort all vertices again based on the negative of the downstream distance of their most-downstream neighbor. The final step of the algorithm (lines 22-27) combine the two permutation vectors (P_1 and P_2) into the final permutation vector, P . Note that the vector of downstream locations to be sorted by the second sort is ordered based on the permutation vector determined from the first sort (P_1).

The effect of this sorting algorithm is evident in Fig. 6.4. Figure 6.4 represents the AIC matrix for the same mesh as Fig. 6.2b but with the sorting algorithm applied. After sorting is applied, the system matrix is upper-pentagonal and has a small bandwidth.

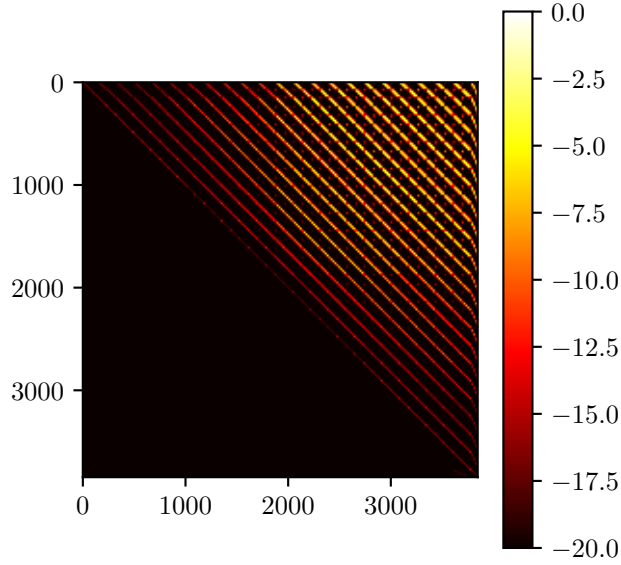


Fig. 6.4: Heatmap representing the AIC matrix coefficients for the wing represented in Fig. 6.2b after sorting the mesh vertices using Algorithm 1.

The lower bandwidth of an upper-pentagonal matrix may be found by looping through each row of the matrix to find the greatest distance between the diagonal and the left-most non-zero element in the row. This procedure is described in Algorithm 2.

As stated previously, if higher-order distributions are used, then more vertices may exert an upstream influence on a given point than with lower-order distributions. The influence of a vertex extends over all panels that vertex touches as well as all panels abutting those panels, except for where there is a discontinuous edge between the two panels. These extra vertices are shown as the yellow vertices in Fig. 6.5. This has the effect of increasing the lower bandwidth of the AIC matrix once sorted. To account for this, the above sorting algorithm must be modified somewhat when higher-order distributions are used. Another do loop would be added where the vertices for these abutting panels that have not already been considered are also considered. This would be part of the computation on line 16 of Algorithm 1.

Algorithm 2 Method for determining the lower bandwidth of a given matrix A

```

1: Input
2:    $A$    System matrix
3: Output
4:    $B_l$   Lower bandwidth
5:  $B_l \leftarrow 0$ 
6:  $i \leftarrow N$ 
7: while  $i > 0$  do                                 $\triangleright$  Loop through rows of the matrix starting at the bottom
8:   found_nonzero  $\leftarrow$  False
9:    $j \leftarrow 1$ 
10:  while  $j < i$  and not found_nonzero do           $\triangleright$  Loop through columns from the left to
the diagonal
11:    if  $|A_{ij}| > 1e - 12$  then                     $\triangleright$  Check for a nonzero element at the current location
12:      found_nonzero  $\leftarrow$  True
13:    else
14:       $j = j + 1$ 
15:    end if
16:  end while
17:  if found_nonzero then
18:     $B_l \leftarrow \max(B_l, i - j)$                  $\triangleright$  Keep the largest lower bandwidth
19:  end if
20:   $i = i - 1$ 
21: end while

```

6.2 System Solvers

To determine the most efficient and robust method for solving the linear system of equations, several different solvers were considered here.

6.2.1 LU Decomposition

The baseline solver for this study was LU decomposition with partial pivoting. The complexity of this method is $O(N^3)$, making it computationally expensive for large meshes. However, it is theoretically exact for well-conditioned systems and robust [126]. It is also the solver that has typically been used in supersonic panel methods (e.g. see [7, 33]).

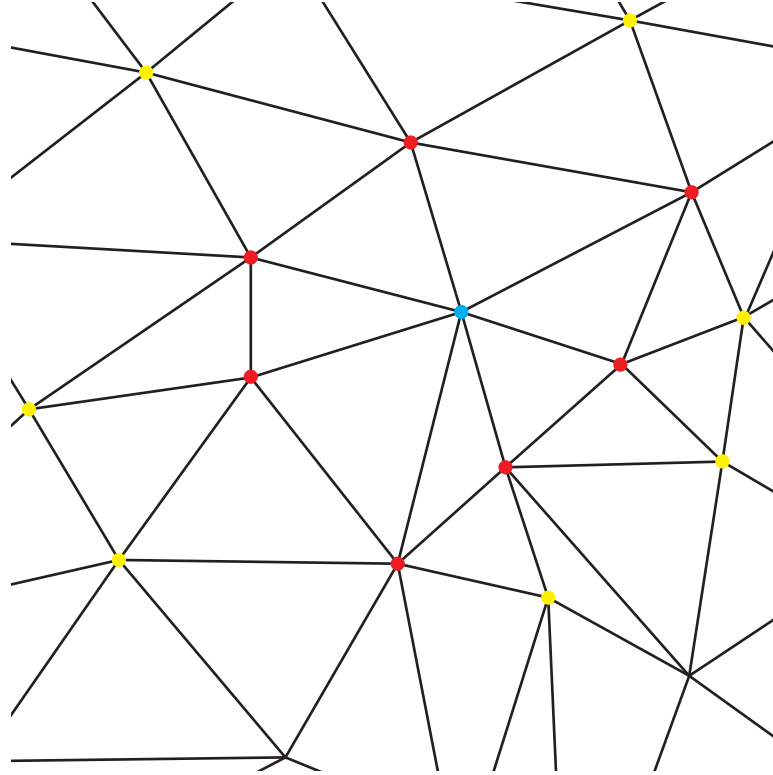


Fig. 6.5: Vertices on an unstructured mesh that must be considered in the second sorting pass for determining the sorting parameter for the blue vertex. Lower-order distributions require considering the red vertices. Higher-order distributions require considering the red and yellow vertices. If the edge shared by both panels touching a yellow vertex and the blue vertex is discontinuous, then that yellow vertex is not considered.

6.2.2 Block-Iterative Solvers

Two basic iterative solvers are considered here: the block symmetric successive overrelaxation [128] and block Jacobi methods [129]. These methods are attractive mainly for their simplicity. They have also been shown previously to work well for the dense, asymmetric systems arising from elliptic BEMs (e.g. see [95–97]).

For both subsonic and supersonic flow, the AIC matrix is asymmetric and indefinite, for which standard successive overrelaxation and Jacobi iterations are not guaranteed to converge. To alleviate this, both are relaxed, and the block-iterative versions are used rather than the traditional, point-iterative versions. To save time, the blocks along the diagonal are decomposed (using LU decomposition) before the iterations begin.

6.2.3 Generalized Minimum Residual Solver

An alternative iterative method is the generalized minimum residual (GMRES) algorithm [130]. This method has become widely used within modern panel methods (and boundary element methods in general) due to its speed and ability to handle non-symmetric and non-definite matrices (e.g. see [99–104, 131]). GMRES is a Krylov subspace method, meaning it iteratively builds an approximation to the solution \mathbf{x} in the space

$$\mathcal{K} = \text{span} \{r_0, Ar_0, A^2r_0 \dots\} \quad (6.3)$$

where $r_0 = Ax_0 - b$ for some initial guess x_0 . Each iteration of the GMRES algorithm involves adding another dimension to \mathcal{K} , updating the orthogonal basis for \mathcal{K} , and then determining the optimal solution in the expanded basis.

One of the disadvantages of GMRES is that its memory and computation requirements steadily grow with each iteration [99]. However, it is guaranteed to converge in N iterations for an $N \times N$ system, though convergence is typically much faster, making GMRES close to $O(N^2)$ [100]. The GMRES algorithm used here was implemented based on the description given in [130].

6.2.4 Novel QR Decomposition Solvers

Another direct method for solving linear systems of equations is the QR decomposition, which factors a matrix A into the form

$$A = QR \quad (6.4)$$

where Q is an orthogonal matrix (i.e. $Q^T = Q^{-1}$) and R is upper triangular [126]. Once A has been decomposed, the equation $Ax = b$ may be solved quickly, as it may be written

$$Rx = Q^T b \quad (6.5)$$

Since R is upper-triangular, the system of equations represented in Eq. (6.5) may be solved using back-substitution in $O(N^2)$ time. In actual implementation, the product $Q^T b$ is usually calculated instead of calculating Q by itself to save time.

The QR factorization can be obtained using Givens rotations [126,132]. In this method, successive rotations are applied to A that zero out each of the elements of A below the diagonal, beginning at the bottom left and proceeding up each column successively. Thus, A is gradually replaced with R . At the same time, these rotations are applied to b , building up $Q^T b$ in place of b .

The algorithm for a single Givens rotation that will zero out an element of A is given in Algorithm 3. Each nonzero subdiagonal element is zeroed out using the diagonal element in the same column. This saves significant time over performing rotations between adjacent matrix elements, as the latter option may result in a large number of trivial swaps with elements that are already zero (e.g. in the case where a column contains a single non-zero element several rows below the diagonal element). As each element is zeroed out, the rotation is also applied to the rest of those two rows and the corresponding elements of b using Algorithm 4.

Algorithm 3 Method for generating a standard Givens rotation to zero out element A_{ij} using element x .

```

1: Input
2:    $A_{jj}$  Matrix element (on the diagonal) used to zero out  $A_{ij}$ 
3:    $A_{ij}$  Matrix element to be zeroed out
4: Output
5:    $c$  Rotation cosine
6:    $s$  Rotation sine
7:  $d \leftarrow \sqrt{A_{jj}^2 + A_{ij}^2}$ 
8:  $c \leftarrow A_{jj}/d$ 
9:  $s \leftarrow A_{ij}/d$ 
10:  $A_{jj} \leftarrow d$ 
11:  $A_{ij} \leftarrow 0$ 

```

Applying a single Givens rotation to the necessary rows of A and b takes $O(N)$ time. Using Givens rotations to reduce a dense matrix to upper-triangular form then takes $O(N^3)$

Algorithm 4 Method for applying a Givens rotation to a pair of row vectors.

```

1: Input
2:    $x$    Upper row vector
3:    $y$    Lower row vector
4:    $c$    Rotation cosine
5:    $s$    Rotation sine
6:  $t \leftarrow cx + sy$ 
7:  $y \leftarrow cy - sx$ 
8:  $x \leftarrow t$ 

```

time, since there are $O(N^2)$ elements to be eliminated. However, R may be obtained faster if A has a large number of zero elements and it is known which elements are already zero. For example, Givens rotations are often used to reduce a Hessenberg matrix (which has zero elements everywhere below the first subdiagonal) to an upper-triangular matrix [133]. Since there are only $O(N)$ nonzero elements below the diagonal of a Hessenberg matrix, the reduction to upper-triangular form using Givens rotations takes only $O(N^2)$ time. Note that a Hessenberg matrix is an upper-pentagonal matrix with lower bandwidth $B_l = 1$. For an upper-pentagonal matrix with lower bandwidth $B_l = 2$, reduction to upper-triangular form will take only $O(2N^2)$ time. Extrapolating this, any given upper-pentagonal matrix with lower bandwidth B_l may be reduced to upper-triangular form in $O(B_l N^2)$ time. If $B_l \ll N$, then such a method should be much faster than $O(N^3)$ solvers such as LU decomposition. Hence the focus previously on permuting the linear system to obtain an upper-pentagonal matrix with small lower bandwidth

A method for solving an upper-pentagonal system using Givens rotations is described in Algorithm 5. This method is called the QR upper-pentagonal (QRUP) solver. This algorithm computes R and $Q^T b$ for an upper-pentagonal system, taking advantage of the known sparsity of A , and then solves for x using back substitution.

Since the time complexity of the QRUP solver is proportional to the lower bandwidth of the AIC matrix, it can be anticipated that the use of higher-order distributions would slow down the QRUP solver relative to if lower-order distributions were used. This effect will be examined subsequently.

Algorithm 5 QRUP solver.

```

1: Input
2:    $N$    System dimension (system is assumed square)
3:    $A$    System matrix
4:    $b$    RHS vector
5: Output
6:    $x$    Solution vector.
7:  $B_l \leftarrow$  output from Algorithm 2
8:  $j \leftarrow 1$ 
9: while  $j \leq N$  do                                      $\triangleright$  Loop through columns
10:    $i \leftarrow \min(j + B_l, N)$                               $\triangleright$  Start at bottom of lower band
11:   while  $i > j$  do                                        $\triangleright$  Loop up through the rows to the diagonal
12:     if  $A_{ij} \neq 0$  then                                    $\triangleright$  If  $A_{ij} = 0$ , no rotation is necessary
13:        $A_{jj}, A_{ij}, c, s \leftarrow$  Generate Givens rotation to zero out  $A_{ij}$  using  $A_{jj}$ 
14:       Apply Givens rotation to the rest of rows  $j$  and  $i$  of  $A$  using  $c$  and  $s$ 
15:       Apply Givens rotation to rows  $j$  and  $i$  of  $b$  using  $c$  and  $s$ 
16:     end if
17:      $i \leftarrow i - 1$ 
18:   end while
19:    $j \leftarrow j + 1$ 
20: end while
21:  $x \leftarrow$  solution of  $Rx = Q^T b$  from back substitution

```

One disadvantage to standard Givens rotations is that each rotation requires computing a square root (Line 7 in Algorithm 3), which can be computationally expensive [126]. However, fast Givens rotations have been formulated that do not compute square roots [1, 133–135]. These fast rotations avoid square roots by factoring the matrix A into the form

$$A = DY \tag{6.6}$$

where D is a diagonal matrix. The squares of the denominators in the standard Givens rotations are stored in D . The fast Givens rotations as originally formulated in [133] are impractical, as D and Y must be monitored and periodically normalized to prevent overflow, negating the intended speed benefit [134].

Later, Anda and Park presented an alternative formulation for fast Givens rotations [1, 135] that dynamically scales D to prevent overflow. The specific form of the rotation is

dependent upon the relative magnitudes of the elements being rotated (A_{jj} and A_{ij}) and the diagonal elements in the corresponding rows (D_j and D_i). The formulas for these fast rotations are given in Table 6.1. For all four rotation types, the final step of applying the rotation is setting A_{ij} to zero.

Table 6.1: Formulas for generating a fast Givens rotation, taken from [1]. Note, as the final step in each case, A_{ij} must be set to zero.

	$ D_j \geq D_i $	$ D_j < D_i $
$D_j/D_i \geq A_{ij}^2/A_{jj}^2$	$rt \leftarrow 1$ $\tau \leftarrow A_{ij}/A_{jj}$ $\beta \leftarrow \tau/\gamma$ $\delta \leftarrow 1 + \beta\tau$ $\alpha \leftarrow \tau/\delta$ $D_j \leftarrow D_j/\delta$ $D_i \leftarrow D_i\delta$ $A_{jj} \leftarrow A_{jj}\delta$	$rt \leftarrow 2$ $\alpha \leftarrow A_{ij}/A_{jj}$ $\tau \leftarrow \alpha/\gamma$ $\delta \leftarrow 1 + \alpha\tau$ $\beta \leftarrow \tau/\delta$ $D_j \leftarrow D_j\delta$ $D_i \leftarrow D_i/\delta$ $A_{jj} \leftarrow A_{jj}$
$D_j/D_i < A_{ij}^2/A_{jj}^2$	$rt \leftarrow 3$ $\alpha \leftarrow A_{jj}/A_{ij}$ $\tau \leftarrow \alpha\gamma$ $\delta \leftarrow 1 + \alpha\tau$ $\beta \leftarrow \tau/\delta$ $t \leftarrow D_i\delta$ $D_i \leftarrow D_j/\delta$ $D_j \leftarrow t$ $A_{jj} \leftarrow A_{ij}$	$rt \leftarrow 4$ $\tau \leftarrow A_{jj}/A_{ij}$ $\beta \leftarrow \tau\gamma$ $\delta \leftarrow 1 + \beta\tau$ $\alpha \leftarrow \tau/\delta$ $t \leftarrow D_i/\delta$ $D_i \leftarrow D_j\delta$ $D_j \leftarrow t$ $A_{jj} \leftarrow A_{ij}\delta$

In order to use Anda and Park's [1] algorithm for the solution of a standard system of equations, the matrix equation must be written as

$$Ax = b \rightarrow DYx = Dc \quad (6.7)$$

In the fast Givens algorithm, D is initialized to the identity matrix, which means, initially, $c = b$ and $Y = A$. At each step of the algorithm, a fast Givens rotation is generated which zeros out an element of Y and updates two elements of D . This rotation is then applied to the other columns of Y and the corresponding rows of c (see Algorithm 6). Once the triangularization is complete, D may be factored out of both sides of Eq. (6.7), resulting in

$$Yx = c \tag{6.8}$$

which may be solved in $O(N^2)$ time using back-substitution, since Y is upper-triangular. Note that D is not used explicitly to calculate x , but it is needed for calculating the fast Givens rotations.

Algorithm 6 Method for applying a fast Givens rotation.

```

1: Input
2:    $x$    Upper row vector
3:    $y$    Lower row vector
4:    $\alpha$  Rotation factor
5:    $\beta$    Rotation factor
6:    $rt$   Rotation type
7: if  $rt = 1$  then
8:    $x \leftarrow x + \beta y$ 
9:    $y \leftarrow -\alpha x + y$ 
10: end if
11: if  $rt = 2$  then
12:    $y \leftarrow -\alpha x + y$ 
13:    $x \leftarrow x + \beta y$ 
14: end if
15: if  $rt = 3$  then
16:    $t \leftarrow y$ 
17:    $y \leftarrow -x + \alpha y$ 
18:    $x \leftarrow t - \beta y$ 
19: end if
20: if  $rt = 4$  then
21:    $t \leftarrow x$ 
22:    $x \leftarrow \beta x + y$ 
23:    $y \leftarrow \alpha x - t$ 
24: end if

```

A method for solving an upper-pentagonal system using fast Givens rotations is described in Algorithm 7. This is called the fast QRUP (FQRUP) solver. Similar to the QRUP solver, the FQRUP solver reduces an upper-pentagonal matrix to upper-triangular form in $O(B_l N^2)$ time, but without calculating square roots. Once the system is in upper-triangular form, it is solved using back-substitution.

Algorithm 7 FQRUP solver.

```

1: Input
2:    $N$    System dimension (system is assumed square)
3:    $A$    System matrix
4:    $b$    RHS vector
5: Output
6:    $x$    Solution vector
7:  $B_l \leftarrow$  output from lower bandwidth calculation
8:  $j \leftarrow 1$ 
9:  $D \leftarrow 1$  ▷  $D$  is a length- $N$  vector of diagonal elements
10: while  $j \leq N$  do ▷ Loop through columns
11:    $i \leftarrow \min(j + B_l, N)$  ▷ Start at bottom of lower band
12:   while  $i > j$  do ▷ Loop up through the rows to the diagonal
13:     if  $A_{ij} \neq 0$  then ▷ If  $A_{ij} = 0$ , no rotation is necessary
14:        $A_{jj}, A_{ij}, \alpha, \beta, D_j, D_i, rt \leftarrow$  Generate fast Givens rotation to zero out  $A_{ij}$  using
        $A_{jj}$ 
15:       Apply fast Givens rotation of type  $rt$  to the rest of rows  $j$  and  $i$  of  $A$  using  $\alpha$ 
       and  $\beta$ 
16:       Apply fast Givens rotation of type  $rt$  to rows  $j$  and  $i$  of  $b$  using  $\alpha$  and  $\beta$ 
17:     end if
18:      $i \leftarrow i - 1$ 
19:   end while
20:    $j \leftarrow j + 1$ 
21: end while
22:  $x \leftarrow$  Solution of  $Yx = c$  from back substitution

```

6.3 Solver Comparison

The relative performance of each of the above solvers was assessed by using them to solve Eq. (2.22) as it appeared for three different configurations. The three configurations considered here were a circular cone with a 10° half angle, a straight, double-wedge wing with a 5° half angle, and a wing-body configuration with flow-through engine nacelles. These configurations are shown in Figs. 6.6-6.8. These different configurations allowed for exploring the effect of variations in the matrix structure on solver performance. For each case, three different mesh densities were analyzed in order to estimate the computational complexity of each solver. The resultant system dimensions for each mesh refinement level and the freestream Mach numbers used are given in Table 6.2 for each configuration.

Each solver was tested with and without sorting the system. While sorting is considered required for the QRUP and FQRUP solvers, initial tests revealed that the sorting algorithm

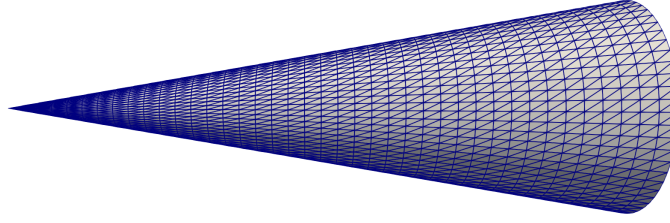
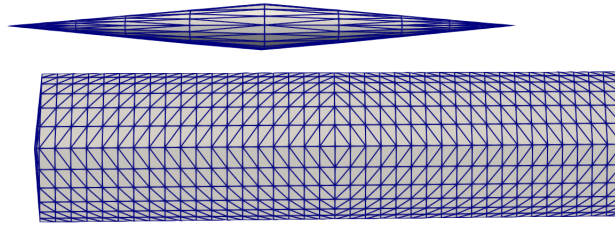
Fig. 6.6: Medium mesh for the 10° half-angle cone.

Fig. 6.7: Medium mesh for the double-wedge wing.

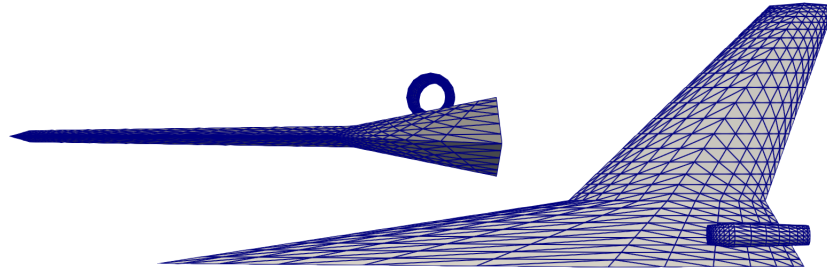


Fig. 6.8: Medium mesh for the wing-body-nacelle combination.

Table 6.2: Freestream Mach numbers and system dimension (N) for the three configurations considered.

Configuration	M_∞	N		
		Coarse	Medium	Fine
Cone	1.5	321	1241	4881
Wing	2.0	419	1127	3847
Wing-Body-Nacelle	2.0	332	1285	5541

also sometimes improved the performance of the iterative methods. Each solver was also tested with and without diagonal preconditioning, as in [99].

A relaxation factor of 0.8 was used for the BJAC and BSSOR solvers, which was found to give a good compromise between speed and convergence for most configurations.

The block size for these was $N/5$, meaning the block size changed with each configuration and mesh refinement. Initial tests showed that this block size provided a good balance between the time it took to calculate the block decompositions and the time it took to run the iterations, though this may be tuned to a specific case, meaning the iterative results presented here may not be representative of the ideal. All iterative solvers had a termination tolerance of 10^{-12} , and a maximum of 1000 iterations were allowed.

For each solver, the total time required to sort the linear system (or not), apply the diagonal preconditioner (or not), and solve the system was recorded and then averaged over five repeated tests. The norm of the final residual vector was recorded (again averaged) to assess the accuracy of each solver. All tests were run on an HP Z240 workstation with an Intel i7 Seventh-Generation processor and 16 GB of RAM. All solvers were implemented in Fortran as part of the MachLine pilot code and compiled using gfortran. Though the efficiency of most of these methods may be increased through parallelization, determining the effect of parallel processing is beyond the scope of the current work, and all results presented here were obtained using only a single thread.

6.3.1 Cone

The first case was a circular cone with a 10° half angle. The slender shape of the cone made it such that the upper-pentagonal AIC matrix was very filled-in (see Fig. 6.3). This was because the domain of influence for any given panel encompassed much of the rest of the cone downstream of it, particularly for panels closer to the tip.

Figures 6.9 and 6.10 show the matrix solver run times for the 10° half-angle cone. For the coarse and medium meshes, the QRUP and FQRUP solvers were fastest out of all solvers considered. However, for the fine mesh, GMRES was the fastest. Sorting the system significantly improved the run times only for the QRUP and FQRUP solvers, while diagonal preconditioning had no noticeable effect on the run time of any solver. Surprisingly, the FQRUP solver was not noticeably faster than the QRUP solver. This is likely because of the square root only being a small part of applying a standard Givens rotation. In addition,

determining the form of the fast rotation for each step of the FQRUP solver requires some extra overhead, decreasing its relative efficiency.

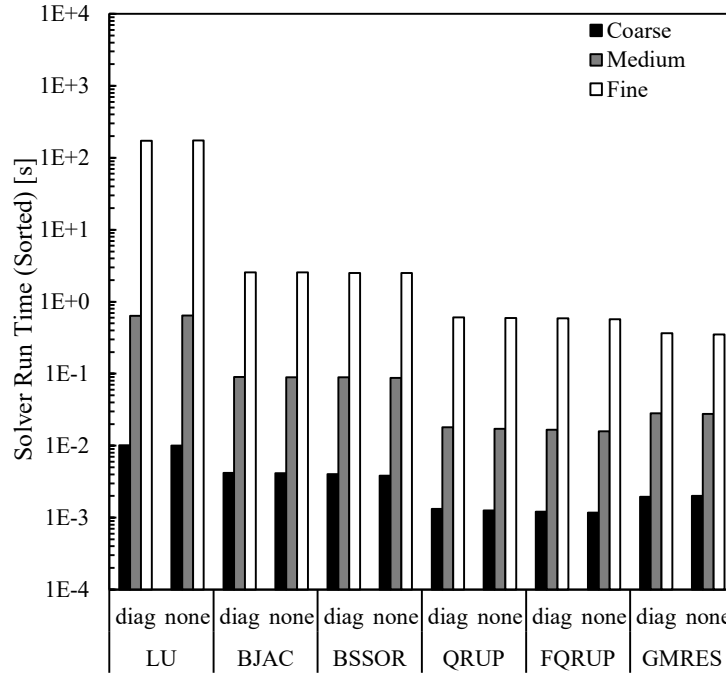


Fig. 6.9: Average run times for the cone case with the system sorted.

Figures 6.11 and 6.12 show the solution residual norms for the different matrix solvers. The QRUP and FQRUP solvers produced acceptable residuals when the system was sorted. However, these produced unacceptably high residuals when the system was not sorted. This may be because the many operations required to reduce a non-upper-pentagonal matrix to upper-triangular form resulted in a non-trivial buildup of numerical error, something not seen when relatively few operations were used on the upper-pentagonal (sorted) system. Preconditioning improved the final accuracy of the QRUP solver, but not the FQRUP solver. This may be due to how the FQRUP solver stores the matrix diagonal separately.

Since the iterative solvers were given the somewhat arbitrary termination tolerance of 10^{-12} , it is useful to consider whether a different termination tolerance would have improved their performance relative to the QRUP or FQRUP solvers. Figures 6.13-6.15

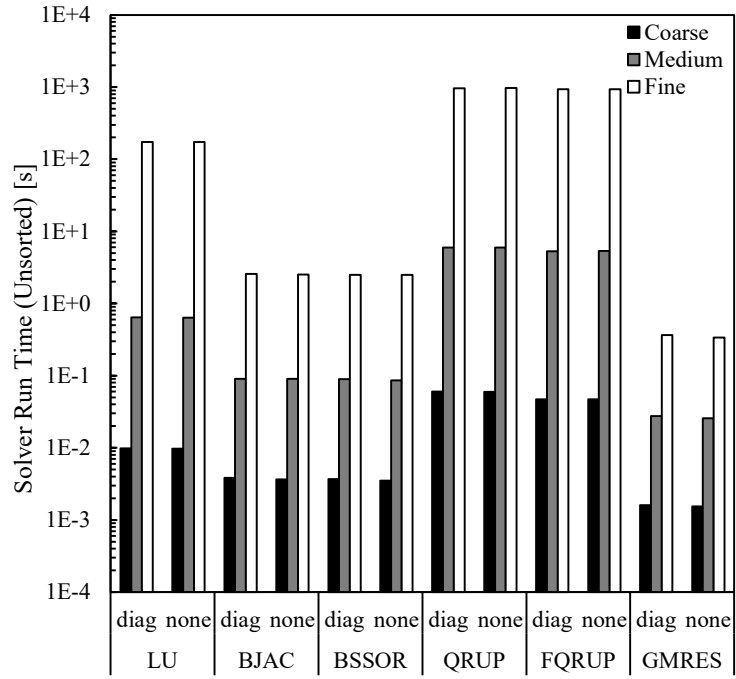


Fig. 6.10: Average run times for the cone case with the system unsorted.

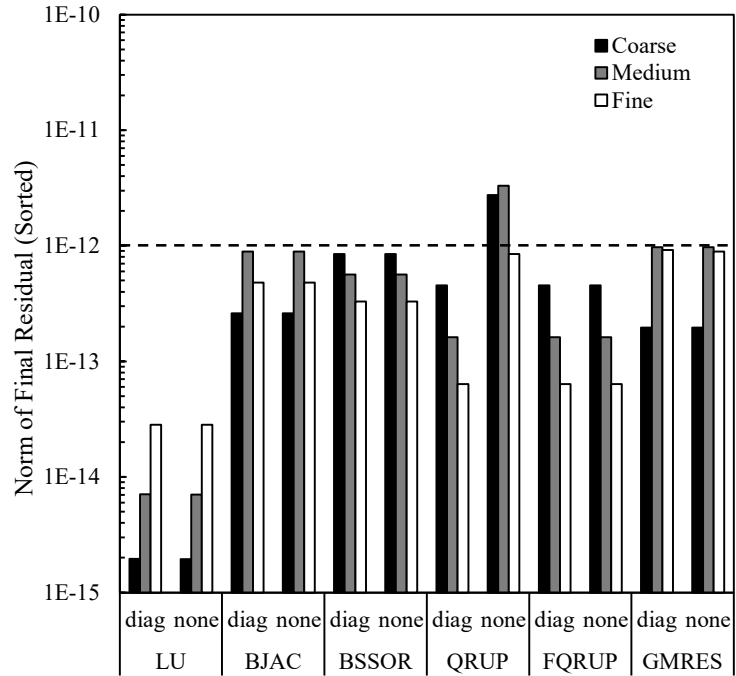


Fig. 6.11: Average residual norms for the cone case with the system sorted. Dashed line indicates the iterative solver termination tolerance.

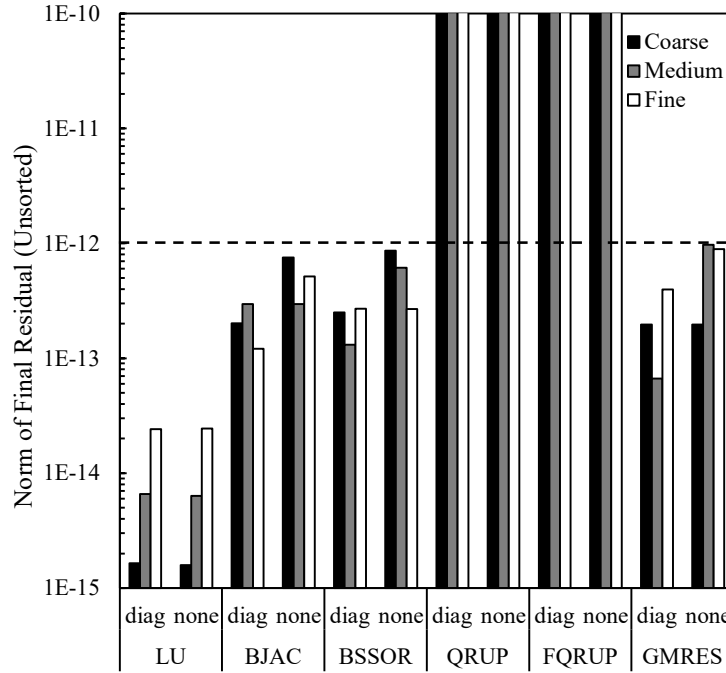


Fig. 6.12: Average residual norms for the cone case with the system unsorted. Dashed line indicates the iterative solver termination tolerance.

show the residual history for each iterative solver as a function of time compared to the final residuals and run times of the QRUP and FQRUP solvers. As plotted here, a solver will be better the closer it is to the bottom-left corner of the plot (i.e. short run time and low final residual). For both the coarse and medium meshes, the QRUP and FQRUP solvers produced a low residual much faster than any of the iterative solvers, with the FQRUP solver being slightly faster and more accurate than the QRUP solver. However for the fine mesh (Fig. 6.15), the general slope of the iteration history for the GMRES solver indicated that it could potentially reach a residual as low as the QRUP and FQRUP solvers faster than either of these. That is, if the termination tolerance for the GMRES solver had been as low as the final residuals from the QRUP and FQRUP solvers, it could have reached this low residual faster.

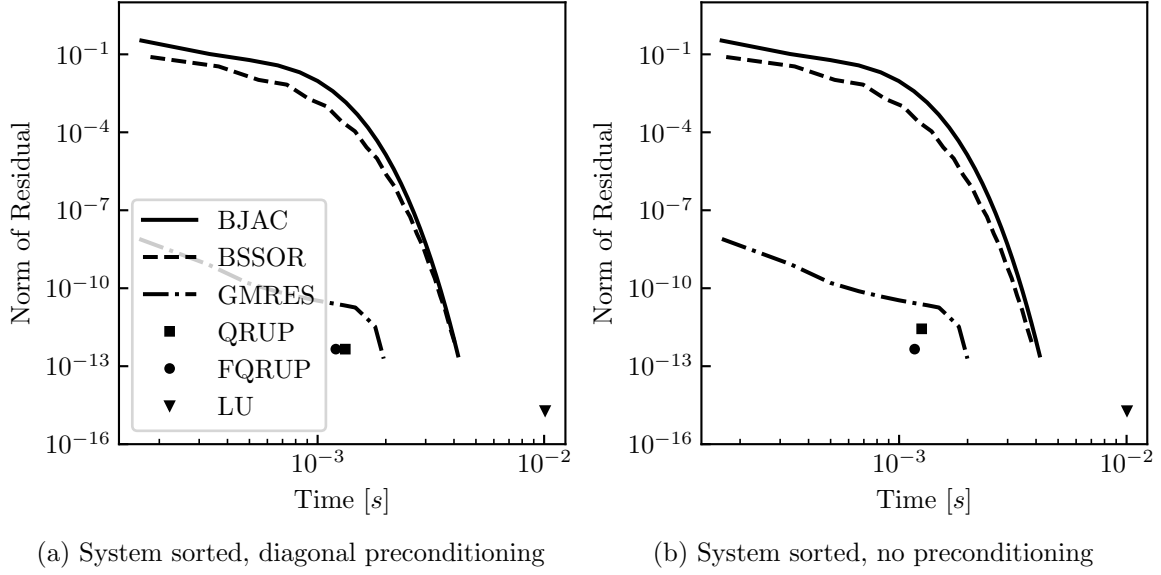


Fig. 6.13: Solution residual versus run time for all solvers considered for the coarse cone mesh.

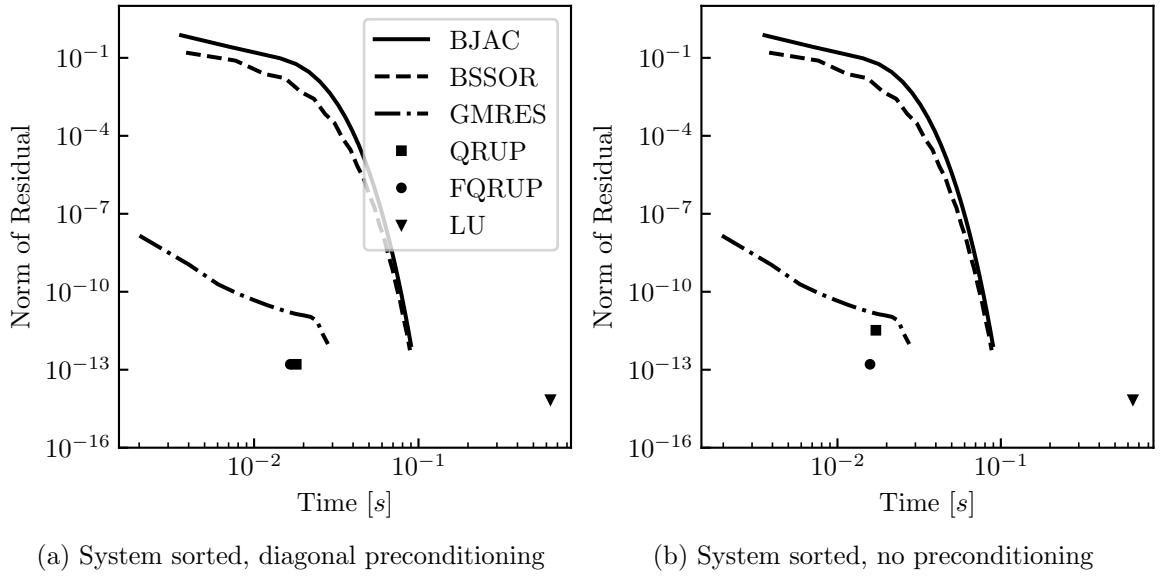


Fig. 6.14: Solution residual versus run time for all solvers considered for the medium cone mesh.

6.3.2 Double-Wedge Wing

The second case considered was a straight, double-wedge wing with a 5° wedge half angle. The relatively high aspect ratio of the double-wedge wing made the upper-pentagonal AIC matrix rather sparse, as shown in Fig. 6.4. This was because any given panel had

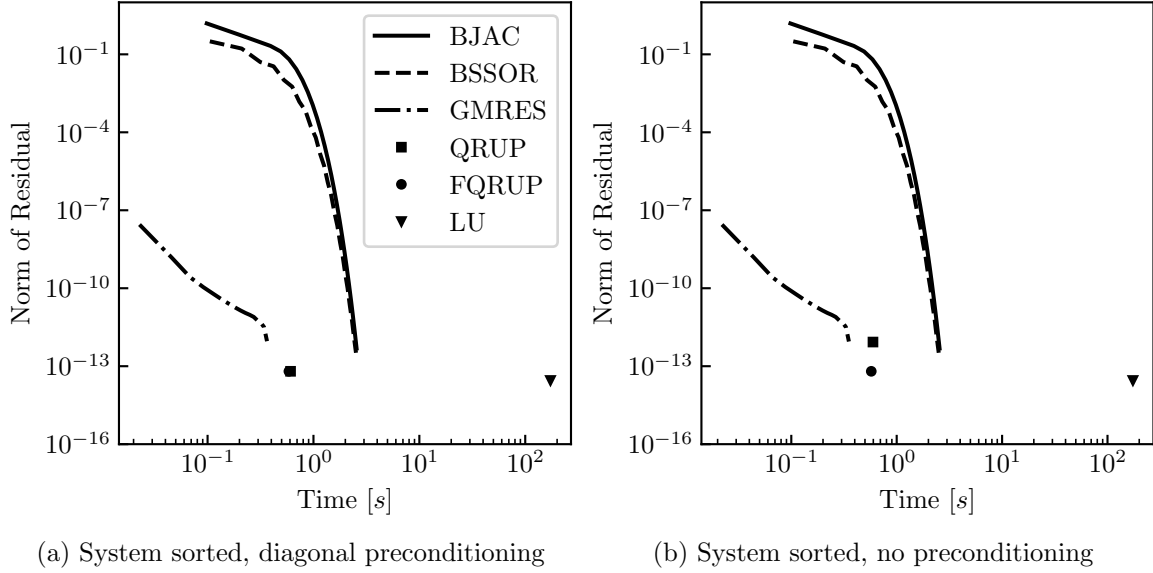


Fig. 6.15: Solution residual versus run time for all solvers considered for the fine cone mesh.

relatively few panels within its domain of influence, even if that panel was near the leading edge.

Figures 6.16 and 6.17 show the solver run times for the double-wedge wing. For this case, the FQRUP solver was fastest for the coarse mesh, while the GMRES solver was the fastest for the medium and fine meshes. As with the cone, not sorting the system significantly increased the run time of the QRUP and FQRUP solvers. In addition, not sorting the system significantly increased the run times for the BJAC and BSSOR solvers as well. Figure 6.2b shows that there were large blocks of nonzero elements far from the diagonal of the unsorted AIC matrix. Thus, sorting the system likely improved diagonal dominance, which is known to improve convergence of the BJAC and BSSOR methods [129].

Figures 6.18 and 6.19 show the residuals for the various solvers. Again, the QRUP and FQRUP solvers produced large residuals without the system sorted. The residuals produced by the QRUP solver were again reduced by diagonal preconditioning, but all other solvers were unaffected. For the sorted system with diagonal preconditioning, all solvers produced acceptably small residuals, though the QRUP and FQRUP solvers did produce noticeably higher residuals than the other solvers.

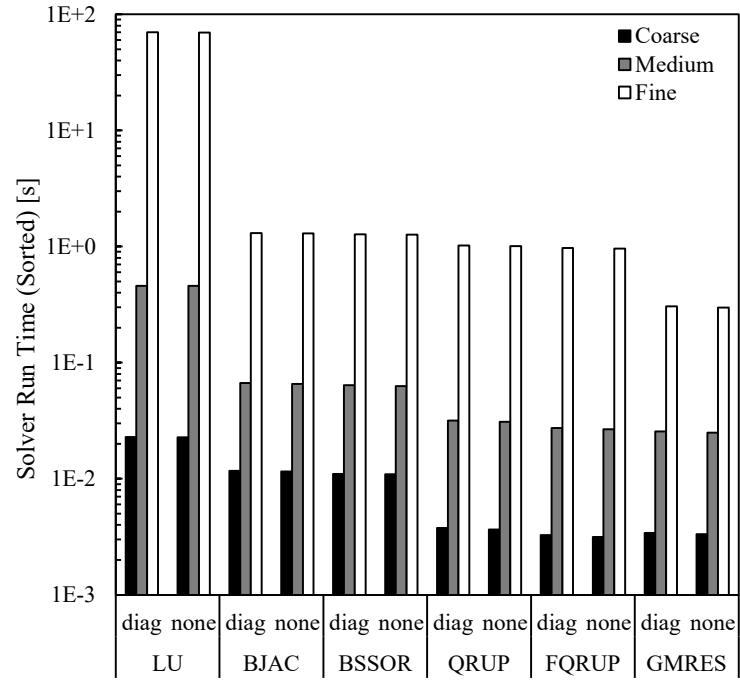


Fig. 6.16: Average run times for the wing case with the system sorted.

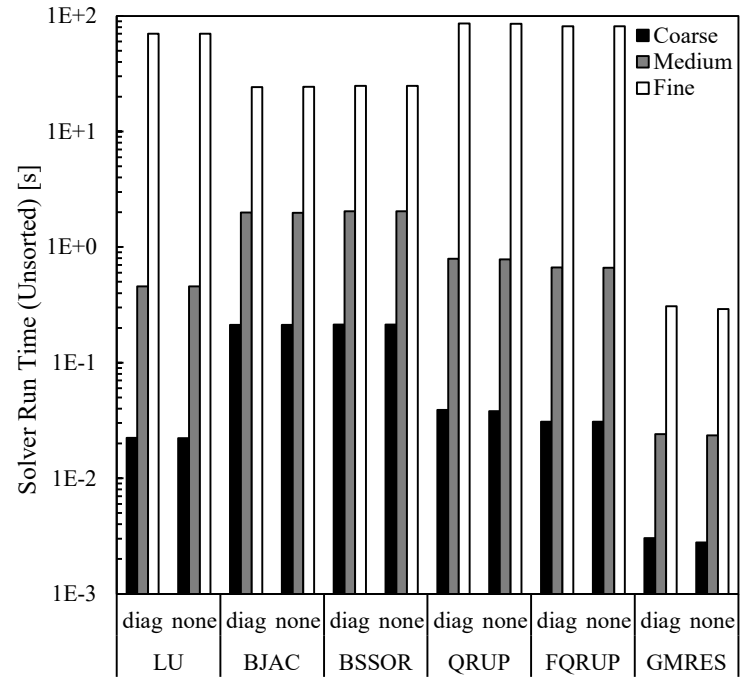


Fig. 6.17: Average run times for the wing case with the system unsorted.

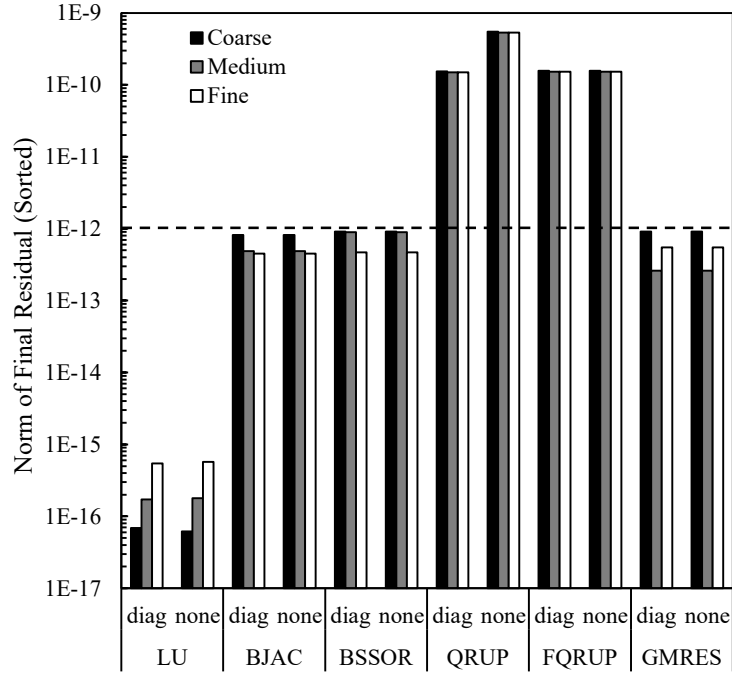


Fig. 6.18: Average residual norms for the wing case with the system sorted. Dashed line indicates the iterative solver termination tolerance.

The residuals versus run time for all solvers for the wing are shown in Figs. 6.20-6.22. Unlike with the cone, the GMRES solver clearly outperformed the QRUP and FQRUP solvers in terms of both accuracy and speed for this case. However, the performance of the QRUP and FQRUP solvers was comparable to that of the BJAC and BSSOR solvers.

6.3.3 Wing-Body-Nacelle Combination

The final configuration considered was a wing-body-nacelle combination. The resultant upper-pentagonal (sorted) AIC matrix for the wing-body-nacelle combination is shown in Fig. 6.23. The structure of this AIC matrix was somewhere in between that of the AIC matrices for the wing and cone. The upper-pentagonal portion was more dense than that of the wing but less dense than that of the cone. The AIC matrix for this case also had a large lower bandwidth, which served to test the efficiency of the QRUP and FQRUP solvers for a non-optimal case.

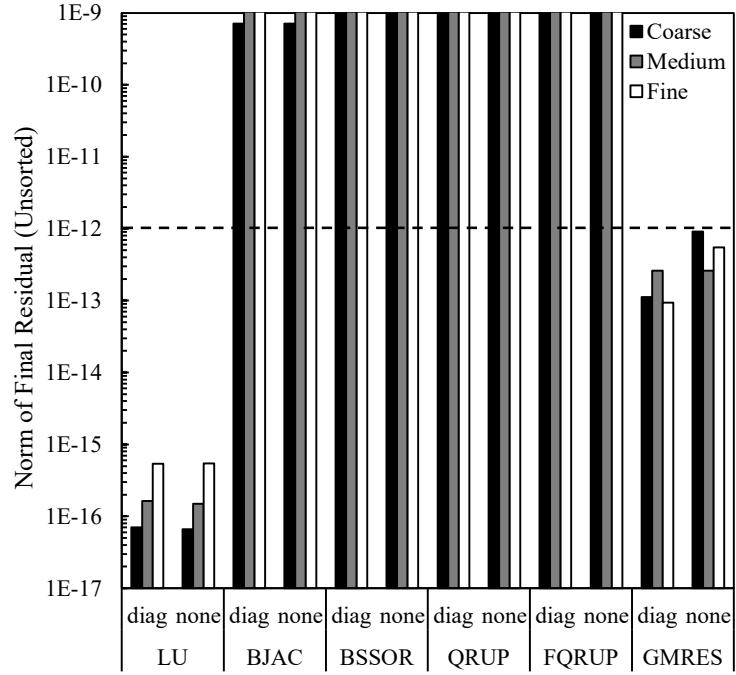
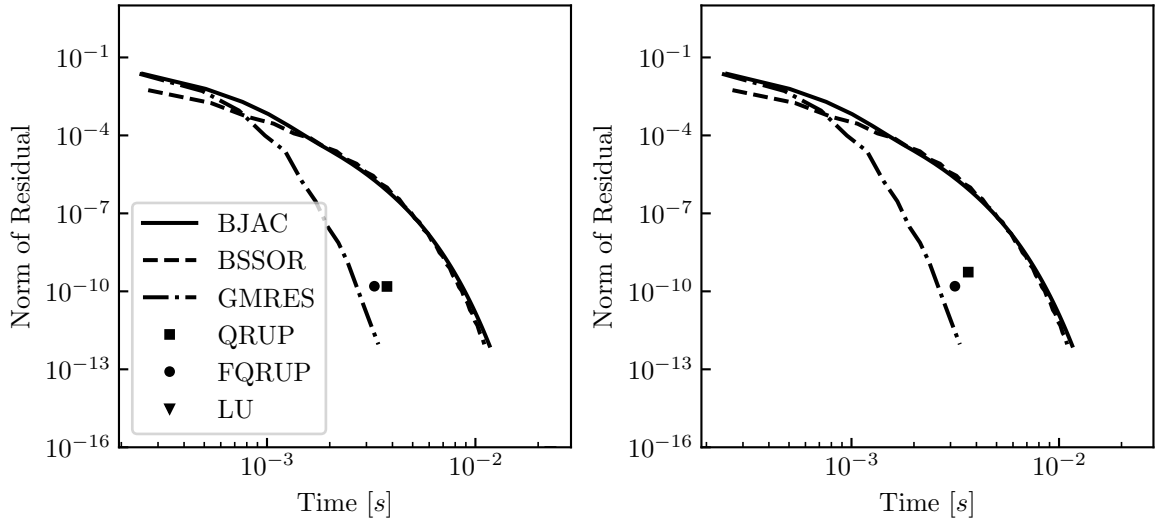


Fig. 6.19: Average residual norms for the wing case with the system unsorted. Dashed line indicates the iterative solver termination tolerance.



(a) System sorted, diagonal preconditioning (b) System sorted, no preconditioning

Fig. 6.20: Solution residual versus run time for all solvers considered for the coarse wing mesh.

Figures 6.24 and 6.25 show the solver run times for the wing-body-nacelle configuration. For this case, both the QRUP and FQRUP solvers were faster than GMRES for the sorted

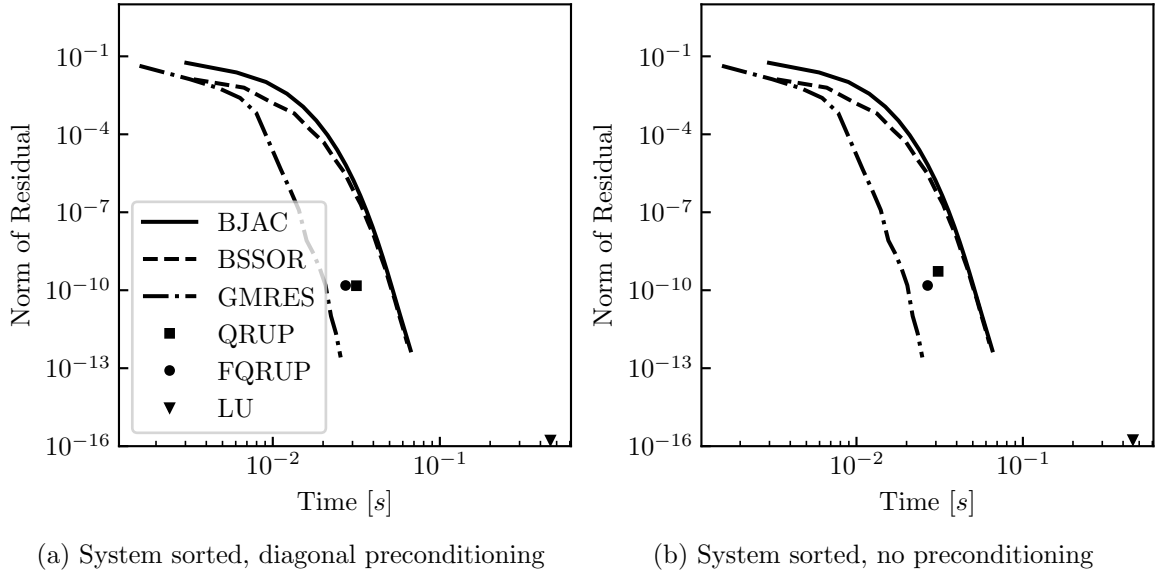


Fig. 6.21: Solution residual versus run time for all solvers considered for the medium wing mesh.

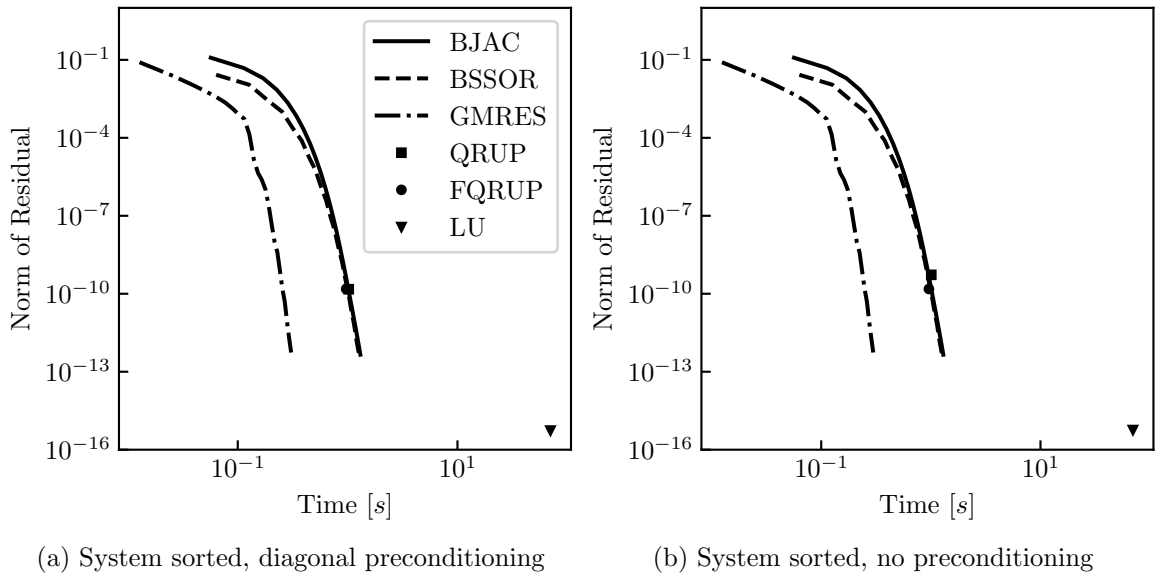


Fig. 6.22: Solution residual versus run time for all solvers considered for the fine wing mesh.

system. However, for the fine mesh, the BJAC and BSSOR solvers both outperformed all other methods, with BSSOR being slightly faster. Interestingly, LU decomposition was faster than all of the iterative solvers for the coarse and medium meshes when the system

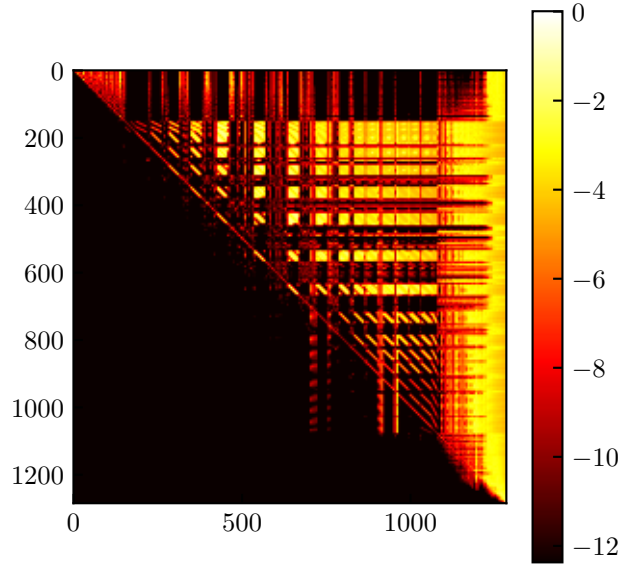


Fig. 6.23: Heatmap representing the AIC matrix for the medium wing-body-nacelle configuration mesh at $M_\infty = 2$.

was sorted. As with the double-wedge wing, sorting the system improved the speed of the BJAC, BSSOR, QRUP, and FQRUP solvers.

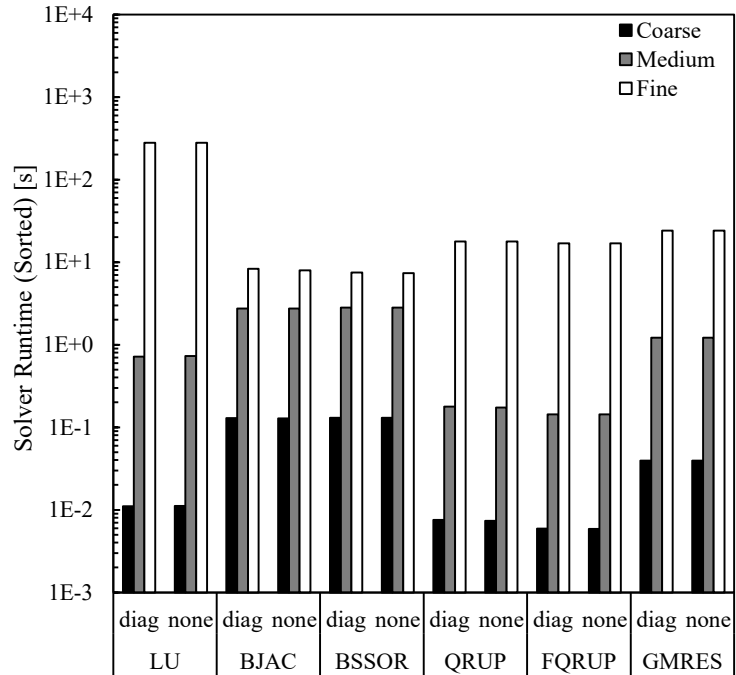


Fig. 6.24: Average run times for the wing-body-nacelle case with the system sorted.

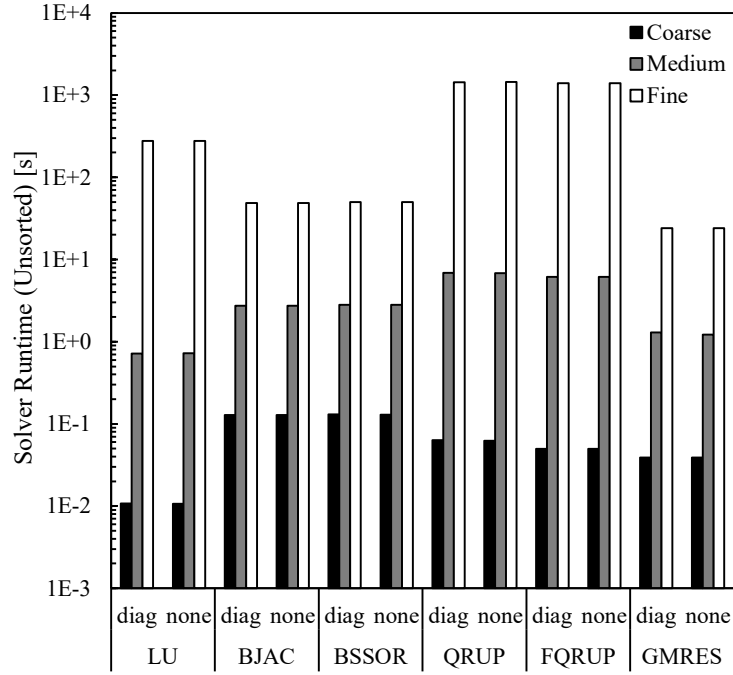


Fig. 6.25: Average run times for the wing-body-nacelle case with the system unsorted.

The final residuals for the wing-body-nacelle configuration are shown in Figs. 6.26 and 6.27. When the system was sorted, the BJAC and BSSOR solvers did not converge for the coarse and medium meshes, and the GMRES solver did not converge for the fine mesh. Without sorting, the BJAC and BSSOR solvers failed to converge for all mesh densities. The residuals produced by the QRUP and FQRUP solvers were very good when the system was sorted, with diagonal preconditioning again resulting in lower residuals from the QRUP solver.

The residual versus time comparison for the wing-body-nacelle configuration is shown for all solvers except LU decomposition in Figs. 6.28-6.30. These comparisons revealed some very interesting behavior. For the coarse mesh, both the BJAC and BSSOR solvers stalled at a residual of about 10^{-7} , indicating that more solution time would not have improved their performance. For this mesh, the QRUP and FQRUP solvers clearly performed better than all the iterative solvers.

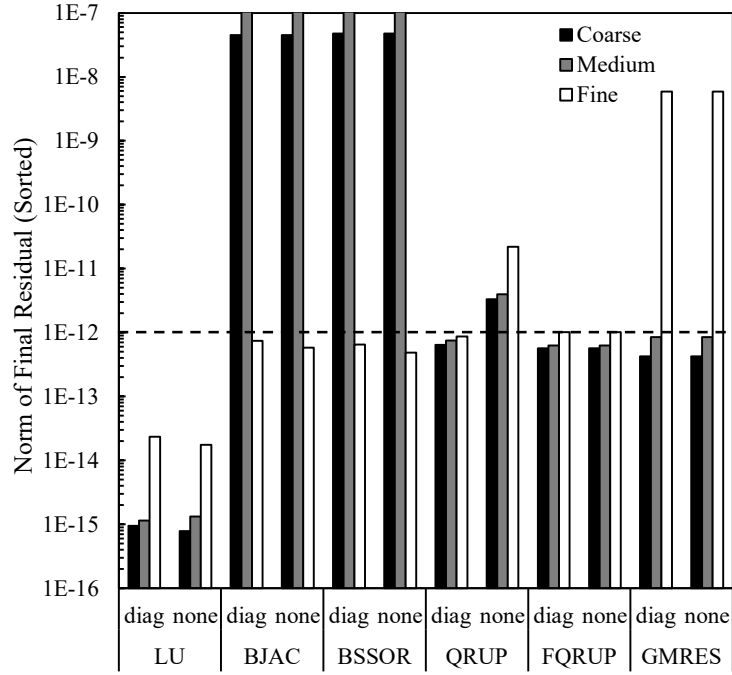


Fig. 6.26: Average residual norms for the wing-body-nacelle case with the system sorted. Dashed line indicates the iterative solver termination tolerance.

For the medium mesh (Fig. 6.29), the BJAC and BSSOR solvers diverged very quickly. And as before, the QRUP and FQRUP solvers significantly outperformed GMRES.

However for the fine mesh, both the BJAC and BSSOR solvers performed better than the QRUP and FQRUP solvers. Though their convergence slowed somewhat around a residual of about 10^{-10} , the residuals generally continued to decrease as the solvers progressed. Thus, given a lower termination tolerance, these solvers could have likely produced more accurate results than either the QRUP or FQRUP solvers. The GMRES solver stopped converging quickly around a residual of 10^{-7} and failed to converge below the given termination tolerance. Because of this, the GMRES solver was significantly outperformed by the QRUP and FQRUP solvers.

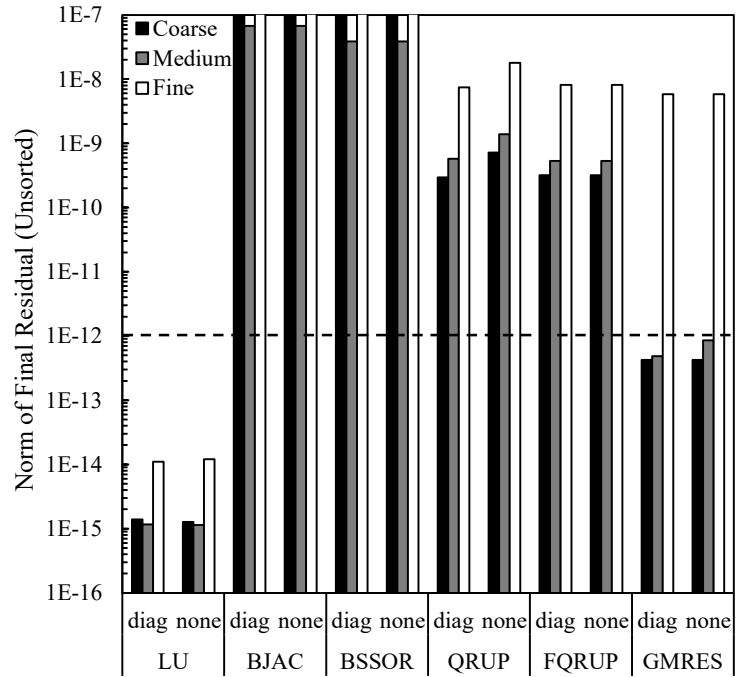
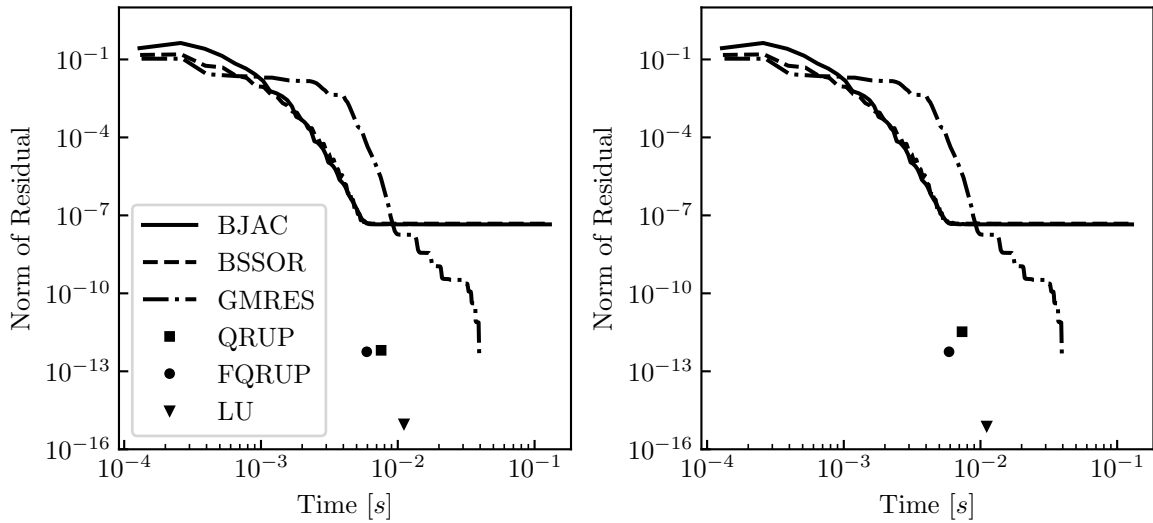


Fig. 6.27: Average residual norms for the wing-body-nacelle case with the system unsorted. Dashed line indicates the iterative solver termination tolerance.



(a) System sorted, diagonal preconditioning

(b) System sorted, no preconditioning

Fig. 6.28: Solution residual versus run time for all solvers considered for the coarse wing-body-nacelle mesh.

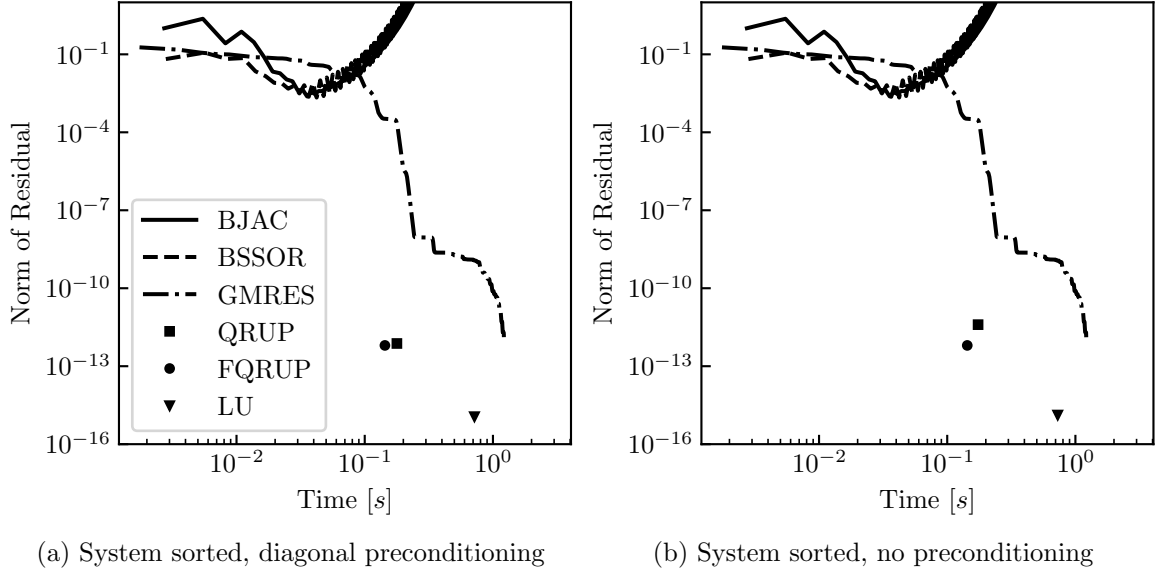


Fig. 6.29: Solution residual versus run time for all solvers considered for the medium wing-body-nacelle mesh.

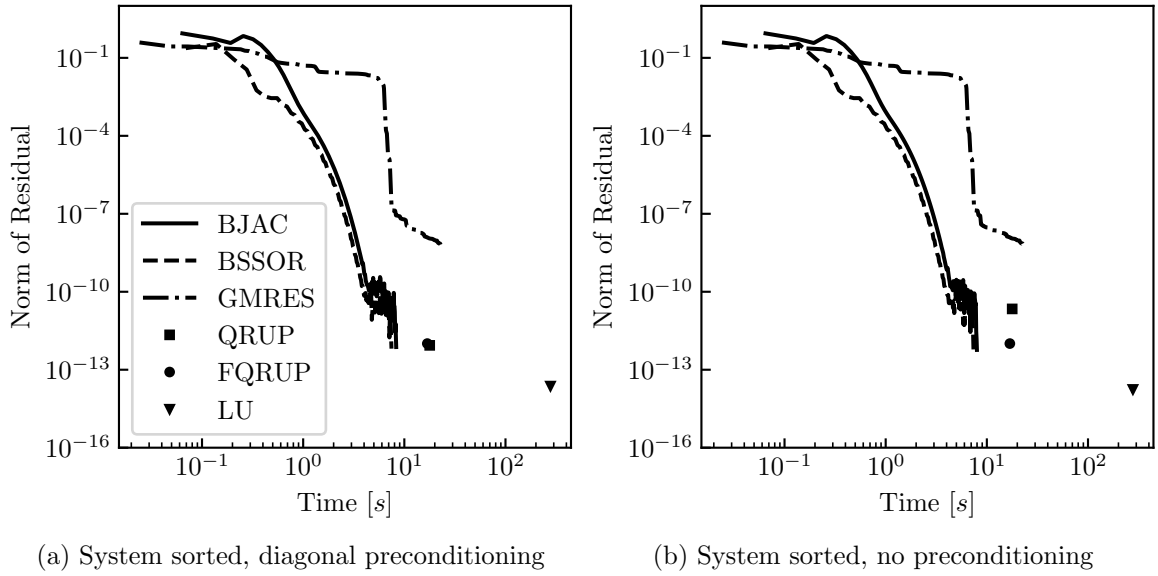


Fig. 6.30: Solution residual versus run time for all solvers considered for the fine wing-body-nacelle mesh.

6.3.4 Time Complexities

The time complexity of a method is often used to predict its performance for general cases. Time complexity is estimated by fitting a power law to the execution time as a

function of the size of the linear system. This was done here for each solver with the system sorted, as this typically produced the best results.

These results averaged across the three configurations are shown in Fig. 6.31. Not surprisingly, the LU solver showed the least variation in complexity between the different configurations, as well as the highest complexity in general. Diagonal preconditioning had little to no effect on the order of complexity of any solver. In all cases, the iterative solvers had lower time complexity than the direct solvers, though the QRUP and FQRUP solvers still showed a significant improvement over the LU solver. The time complexities of the QRUP and FQRUP solvers had significant variation between cases, showing the sensitivity of these solvers to the lower bandwidth (for the cone case, the time complexity was estimated to be 2.3, whereas it was 2.7 for the wing-body-nacelle combination). Out of all the solvers considered, the BJAC and BSSOR solvers had the lowest time complexity. However, these also showed significant variation in complexity, whereas the complexity of GMRES was only slightly higher and varied less between the cases.

6.3.5 Higher-Order Distributions and Lower Bandwidth

As discussed previously, the use of higher-order singularity distributions results in an increased lower bandwidth for the sorted AIC matrix. Exactly how much the lower bandwidth increases depends on the case. For the three configurations examined above, the lower bandwidths for each mesh density resulting from both lower- and higher-order distributions are given in Table 6.3. On average, the use of higher-order distributions increased the lower bandwidth of the AIC matrix by a factor of 1.65, with the smallest increase (a factor of 1.35) occurring for the fine mesh for the wing-body-nacelle combination, and the largest increase (a factor of 1.97) occurring for the fine mesh for the cone. Based on this analysis, it is anticipated that the use of higher-order distributions should increase the time taken to solve the linear system of equations using the QRUP or FQRUP solvers by a factor of approximately 1.65.

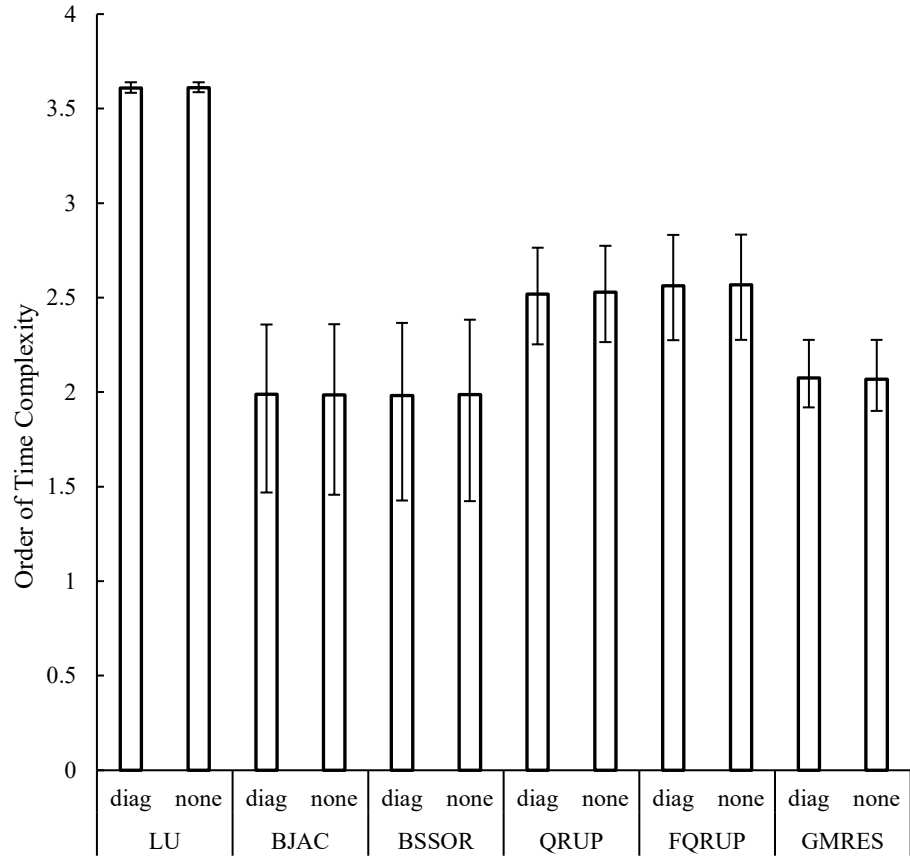


Fig. 6.31: Solver time complexities averaged across the three configurations with and without diagonal preconditioning. The uncertainty bands shown represent the standard deviation between the three configurations.

Table 6.3: Lower bandwidths for the sorted AIC matrix as a function of configuration, system dimension, and the type of singularity distribution used.

Configuration	System Dimension	Lower Bandwidth	
		Higher-Order	Lower-Order
Cone	321	30	16
	1241	60	31
	4881	120	61
Double-Wedge Wing	419	79	52
	1127	139	92
	3847	259	172
Wing-Body-Nacelle	332	204	125
	1285	562	368
	5541	1517	1127

6.4 Discussion

Overall, LU decomposition with partial pivoting was most reliable at accurately solving the system of equations and was largely unaffected by system sorting and preconditioning. However, LU decomposition was consistently the slowest and had a measured order of computational complexity greater than 3.5.

The BJAC solver was consistently fast and generally reliable. For both the double-wedge wing and wing-body-nacelle cases, the speed of the BJAC solver was improved by sorting the linear system to achieve an upper-pentagonal matrix. However, this solver failed to converge for some meshes in the wing-body-nacelle case. Thus, it may not be appropriate for all cases. However, it is easy to implement and has a low order of complexity, making it an attractive option.

The BSSOR solver had very similar performance to the BJAC solver while being slightly faster for some cases. While not explicitly considered here, one point that makes BSSOR less attractive than BJAC is that BJAC is very easily parallelized.

The novel QRUP and FQRUP solvers, while not being the fastest considered, did provide acceptably accurate solutions to the linear system of equations in all cases, as long as the system was sorted into upper-pentagonal form. With the system sorted, the QRUP and FQRUP solvers ran significantly faster than LU decomposition and about on par with the iterative solvers considered.

The purpose of the FQRUP solver was to reduce computation times relative to the QRUP solver. For some of the cases considered here, the FQRUP solver did run slightly faster than the QRUP solver. However, as shown in Fig. 6.31, the FQRUP solver had a slightly higher time complexity than the QRUP solver. Their time complexities should be the same, since the two methods require the same number of steps for a given matrix system, and so the difference in measured time complexities would likely be eliminated by considering more cases.

The QRUP solver consistently produced lower residuals when diagonal preconditioning was used, but the FQRUP solver was unaffected by preconditioning. Thus, FQRUP may be

considered faster because it achieves the same level of accuracy without requiring diagonal preconditioning.

Though the QRUP and FQRUP solvers are direct, they failed to produce residuals as low as LU decomposition. As mentioned previously, this could likely be improved by pivoting. This is likely because some diagonal elements of the resulting upper-triangular matrix (R in Eq. (6.5)) were often small (on the order of 10^{-7}). Swapping rows or columns to put larger elements on the diagonal, as is typically done with LU decomposition, would result in less numerical error, as the back-substitution step requires dividing by the diagonal elements. However, for efficiency, such a partial pivoting scheme would need to increase the lower bandwidth of the system matrix as little as possible.

The GMRES solver showed the best overall performance in terms of speed, accuracy, and time complexity. It seemed to also be unaffected by diagonal preconditioning and whether the system was upper-pentagonal. This makes GMRES an attractive option for implementation in a general panel method, as separate solvers would not need to be used to maximize performance for both subsonic and supersonic cases.

However, it should be noted that the GMRES solver failed to converge for the fine mesh for the wing-body-nacelle combination. In this case, the QRUP and FQRUP solvers were both faster and more accurate than GMRES. When using an iterative solver, one must always be wary of the possibility of non-convergence. More-robust variants of GMRES do exist which are designed to handle singular or near-singular systems (e.g. see [136]). However, the use of such algorithms is outside the scope of the current work. It is also common to restart the GMRES algorithm after several iterations in order to cut down on computation and memory requirements (e.g. see [99,137]). However, restarting typically reduces the convergence rate of GMRES [137] and did not improve performance over standard GMRES for the cases presented here.

CHAPTER 7

POST-PROCESSING

Of course. Because why would we want dimensions? - Dr. Randy Christensen

Once the source and doublet strengths have been determined, the surface velocities and pressures may be calculated, from which the resultant force and moment coefficients may be found.

7.1 Surface Velocity Calculation

The surface velocities may be calculated as the gradient of total potential on the outer surface of the configuration. This gradient may be calculated analytically for both lower- and higher-order distributions. On the outer surface of the configuration, the total potential is equal to

$$\Phi_s = U(\phi_\infty + \phi_i + \mu) \quad (7.1)$$

the gradient of which gives the surface velocity.

For the Morino formulation, $\phi_i \equiv 0$, and so the surface velocity is given by [33]

$$\mathbf{u}_{s\text{Morino}} = U \left(\hat{\mathbf{c}} + \nabla\mu + \sigma \frac{\mathbf{n}}{\{\mathbf{n}, \mathbf{n}\}} \right) \quad (7.2)$$

A derivation of this formula is given in Appendix N of [33]. The source term is included because source distributions induce a jump in the normal derivative of velocity potential but do not influence the actual value of the potential on the boundary.

For the source-free formulation, the total potential on the surface is

$$\Phi_{s\text{source-free}} = U(\phi_\infty + \phi_i + \mu) = U(\mathbf{r}^T \hat{\mathbf{c}} - \mathbf{r}^T [B]^{-1} \hat{\mathbf{c}} + \mu) \quad (7.3)$$

Taking the gradient of Eq. 7.3 yields

$$\mathbf{u}_{\text{source-free}} = U(\hat{\mathbf{c}} - [B]^{-1}\hat{\mathbf{c}} + \nabla\mu) \quad (7.4)$$

Since sources are not used, there is no jump in the normal derivative of velocity potential.

7.2 Surface Pressure Calculation

Once the surface velocities have been determined, the surface pressures may be calculated. The pressure at any point in an irrotational flow may be calculated from the freestream pressure and the local velocity using Bernoulli's equation (Eq. (1.34)). This may also be manipulated to give the local pressure coefficient as a function of the local velocity and the freestream velocity. There are multiple ways to do so, which will be presented here. The derivation of these formulas may be found in Appendix N of [33].

First, the pressure coefficient is defined as

$$C_P = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty U^2} \quad (7.5)$$

where p is the local pressure and p_∞ is the freestream pressure.

Using the full isentropic form of Bernoulli's equation, the pressure coefficient is given by

$$C_{P_{\text{ise}}} = \frac{2}{\gamma M_\infty^2} \left[\left(1 + \frac{\gamma - 1}{2} \left(1 - \frac{\|\mathbf{u}\|^2}{U^2} \right) M_\infty^2 \right)^{\frac{\gamma}{\gamma - 1}} - 1 \right] \quad (7.6)$$

where γ is the ratio of specific heats (assumed constant). Equation (7.6) is referred to as the isentropic pressure rule.

By expanding Eq. (7.6) using the binomial theorem and neglecting cubic and higher-order terms, the pressure coefficient may be approximated as

$$C_{P_{\text{2nd}}} = \frac{-2u'_{xc}}{U} - \frac{(1 - M_\infty^2)u'^2_{xc} + u'^2_{yc} + u'^2_{zc}}{U^2} \quad (7.7)$$

where u'_{x^c} denotes the x^c component of the perturbation velocity, etc. Equation (7.7) is referred to as the second-order pressure rule. The second-order pressure rule may also be derived from Eq. (1.81).

For slender bodies, it is often assumed that the x^c component of the perturbation velocity is small, meaning its square may be neglected. Applying this assumption to Eq. (7.7) yields the slender-body rule

$$C_{P_{\text{sln}}} = \frac{-2u'_{x^c}}{U} - \frac{u'^2_{y^c} + u'^2_{z^c}}{U^2} \quad (7.8)$$

By dropping all nonlinear terms from Eq. (7.7) or (7.8), the linear pressure rule is obtained

$$C_{P_{\text{lin}}} = \frac{-2u'_{x^c}}{U} \quad (7.9)$$

This rule may also be derived from Eq. (1.82).

For incompressible flows, the familiar incompressible pressure rule is also available [33, 34]

$$C_{P_{\text{inc}}} = 1 - \frac{\|\mathbf{u}\|^2}{U^2} \quad (7.10)$$

The incompressible pressure rule may be considered the limit of Eq. (7.6) as $M_\infty \rightarrow 0$ and $\gamma \rightarrow \infty$.

7.2.1 Vacuum Pressure Coefficient

For compressible flows, there is a theoretical lower limit to the pressure coefficient called the vacuum pressure coefficient. It may be seen from the form of Eq. (7.6) that the isentropic pressure coefficient can only be evaluated if [33]

$$1 + \frac{\gamma - 1}{2} \left(1 - \frac{\|\mathbf{u}\|^2}{U^2} \right) M_\infty^2 \geq 0 \quad (7.11)$$

Otherwise, a negative number raised to the power of $\frac{\gamma}{\gamma-1}$ will give an invalid result. Thus, the minimum isentropic pressure coefficient occurs when this quantity is zero [33]. This minimum isentropic pressure coefficient is called the *vacuum pressure coefficient*, because at this point the flow has accelerated enough that the static pressure is zero. This vacuum pressure coefficient is given by

$$C_{P_{\text{vac}}} = -\frac{2}{\gamma M_{\infty}^2} \quad (7.12)$$

In the pilot code, the vacuum pressure coefficient is used to replace any NaNs which occur in the pressure coefficient calculations.

The other pressure rules have no such hard limit and so may predict unrealistically low pressure coefficients. Again, the vacuum pressure coefficient may be used to replace such calculated pressures.

7.2.2 Stagnation Pressure Coefficient

For all flows, there is a theoretical upper limit to the pressure coefficient. This will occur when the local velocity is zero and is called the stagnation pressure coefficient. Setting $\|\mathbf{u}\| = 0$ in Eq. (7.6) yields

$$C_{P_{\text{stag}}} = \frac{2}{\gamma M_{\infty}^2} \left[\left(1 + \frac{\gamma-1}{2} M_{\infty}^2 \right)^{\frac{\gamma}{\gamma-1}} - 1 \right] \quad (7.13)$$

As with the vacuum pressure coefficient, the stagnation pressure coefficient may be used to replace any erroneously high pressure coefficient results from any of the above rules.

7.2.3 Subsonic Pressure Corrections

Panel methods based on the Prandtl-Glauert equation can directly simulate compressible subsonic flows. However, an alternative to this direct computation exists. Because the subsonic Prandtl-Glauert equation is elliptic, it may be related to Laplace's equation by a simple coordinate scaling [21]. Because of this, compressible subsonic flows may be modeled by first assuming the flow is incompressible, solving the flow, and then correcting

the resultant incompressible pressure coefficients for compressibility. Three such corrections are discussed here: the Prandtl-Glauert correction, the Karman-Tsien correction, and the Laitone correction.

The Prandtl-Glauert correction is the simplest of the three corrections considered here. The Prandtl-Glauert correction is derived from the linearized perturbation velocity potential equation. The compressible pressure coefficient is given in terms of the incompressible pressure coefficient and the freestream Mach number by [34]

$$C_P = \frac{C_{P_{\text{inc}}}}{\beta} = \frac{C_{P_{\text{inc}}}}{\sqrt{1 - M_\infty^2}}. \quad (7.14)$$

The assumptions required to reach the result shown in Eq. 7.14 are small perturbations, two-dimensional flow, and inviscid flow. While the Prandtl-Glauert correction is derived for two-dimensional flow, it is often applied to three-dimensional flows with good results [138].

While the Prandtl-Glauert correction is derived from linearized theory, the Karman-Tsien and Laitone corrections attempt to account for some nonlinearities in the flow [34]. The Karman-Tsien correction is derived by assuming the quantity $(1 - M^2)/\rho^2$ is constant [139]. This then yields a relation between the velocities of the compressible and incompressible fluids that may be used to find the pressure coefficient correction. The correction is given by [34, 139]

$$C_P = \frac{C_{P_{\text{inc}}}}{\sqrt{1 - M_\infty^2} + \frac{C_{P_{\text{inc}}}}{2} \frac{M_\infty^2}{1 + \sqrt{1 - M_\infty^2}}} \quad (7.15)$$

To derive the Laitone correction, the Mach number of the flow is not considered constant, but rather is allowed to vary according to the isentropic relations for fluid flow up to the first order [140]. The Laitone correction is given by [34, 140]

$$C_P = \frac{C_{P_{\text{inc}}}}{\sqrt{1 - M_\infty^2} + \frac{C_{P_{\text{inc}}}}{2} \frac{M_\infty^2 \left[1 + \frac{\gamma-1}{2} M_\infty^2\right]}{\sqrt{1 - M_\infty^2}}} \quad (7.16)$$

7.3 Force and Moment Integration

From a flight mechanics perspective, the desired analysis outputs are often the overall forces and moments acting on a configuration, rather than the pressure distributions. These may be obtained by integrating the pressure distributions over the configuration surface. For simplicity, only the force and moment coefficients (rather than the dimensional forces and moments) acting in each of the global directions are calculated.

7.3.1 Force Integration

The force coefficient is defined by

$$C_F = \frac{F}{\frac{1}{2}\rho_\infty U^2 A_{\text{ref}}} \quad (7.17)$$

where F is the dimensional force and A_{ref} is a user-specified reference area. The total force coefficient vector will be the vector sum of the forces acting on each panel. The fictitious inner flow may be neglected here since it is constant and so has a constant pressure coefficient. Thus, only the force acting on each panel due to exterior pressures needs to be considered.

For a single panel, the force due to the external pressure is

$$\begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix}_i = - \iint_{S_i} C_P \mathbf{n}_i^g dS = -J \mathbf{n}_i^g \iint_{S_i} C_P(\xi, \eta) d\xi d\eta \quad (7.18)$$

where J is the area Jacobian given by Eq. (5.19) and C_x , C_y , and C_z are the x , y , and z components of the force coefficient vector.

For linear-doublet-constant-source panels, the pressure is constant across each panel (since the velocity is constant) and so may be pulled out of the integral. In this case, the force coefficient vector is calculated from

$$\begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} = -\frac{1}{A_{\text{ref}}} \sum_{i=1}^{N_{\text{panels}}} C_{P_i} A_i \mathbf{n}_i^g \quad (7.19)$$

where A_i is the area of each panel. The normal vector in Eq. (7.19) is expressed in global coordinates, meaning the calculated force vector will be in the global frame. Alternatively, the normal vector may be expressed in any other frame (such as the compressibility frame) in order to obtain the force vector in that frame.

For quadratic-doublet-linear-source panels, the velocity across each panel is linearly distributed. Examining the form of Eqs. (7.7) - (7.10), most pressure rules will then result in a quadratic distribution of pressure. The exception to this is the isentropic rule. However, under small perturbations, the difference between the isentropic rule and the second-order rule should be small, and so it is assumed that the isentropic pressure distribution may also be reasonably approximated as quadratic [33].

Defining the vector

$$\mathbf{d} = \begin{bmatrix} 1 & \xi & \eta & \frac{1}{2}\xi^2 & \xi\eta & \frac{1}{2}\eta^2 \end{bmatrix} \quad (7.20)$$

(c.f. Eq. (5.152)), Eq. (7.18) may be written as

$$\begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix}_i = -J\mathbf{n}_i^g \iint_{S_i} \mathbf{d}^T [S_\mu]^{-1} \{C_P\} d\xi d\eta \quad (7.21)$$

where $[S_\mu]$ is given by Eq. (5.130) (but this time written for the panel vertices and edge midpoints), and $\{C_P\}$ is the vector of pressure coefficients calculated at each vertex and edge midpoint. The matrix $[S_\mu]^{-1}$ is used here to determine the six quadratic pressure distribution parameters from the values of the pressure distribution sampled at the panel vertices and edge midpoints, i.e.

$$\begin{pmatrix} C_{P_0} \\ C_{P_\xi} \\ C_{P_\eta} \\ C_{P_{\xi\xi}} \\ C_{P_{\xi\eta}} \\ C_{P_{\eta\eta}} \end{pmatrix} = \begin{pmatrix} 1 & \xi_1 & \eta_1 & \frac{1}{2}\xi_1^2 & \xi_1\eta_1 & \frac{1}{2}\eta_1^2 \\ 1 & \xi_2 & \eta_2 & \frac{1}{2}\xi_2^2 & \xi_2\eta_2 & \frac{1}{2}\eta_2^2 \\ 1 & \xi_3 & \eta_3 & \frac{1}{2}\xi_3^2 & \xi_3\eta_3 & \frac{1}{2}\eta_3^2 \\ 1 & \xi_{m_1} & \eta_{m_1} & \frac{1}{2}\xi_{m_1}^2 & \xi_{m_1}\eta_{m_1} & \frac{1}{2}\eta_{m_1}^2 \\ 1 & \xi_{m_2} & \eta_{m_2} & \frac{1}{2}\xi_{m_2}^2 & \xi_{m_2}\eta_{m_2} & \frac{1}{2}\eta_{m_2}^2 \\ 1 & \xi_{m_3} & \eta_{m_3} & \frac{1}{2}\xi_{m_3}^2 & \xi_{m_3}\eta_{m_3} & \frac{1}{2}\eta_{m_3}^2 \end{pmatrix}^{-1} \{C_P\} \quad (7.22)$$

$$\{C_P\} = \begin{pmatrix} C_{P_1} \\ C_{P_2} \\ C_{P_3} \\ C_{P_{m_1}} \\ C_{P_{m_2}} \\ C_{P_{m_3}} \end{pmatrix} \quad (7.23)$$

where C_{P_i} is the pressure at vertex i and $C_{P_{m_i}}$ is the pressure at the midpoint of edge i .

Recognizing $[S_\mu]^{-1}$ and $\{C_P\}$ are constant with respect to the integration, this becomes

$$\begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix}_i = -J\mathbf{n}_i^g \left(\iint_{S_i} \mathbf{d}^T d\xi d\eta \right) [S_\mu]^{-1} \{C_P\} \quad (7.24)$$

The only term not yet determined in this equation is

$$\iint_{S_i} \mathbf{d}^T d\xi d\eta \quad (7.25)$$

Following the example of the PAN AIR developers [33], let

$$C_{ij} = \iint_{S_i} \xi^i \eta^j d\xi d\eta \quad (7.26)$$

from which it may be written

$$\iint_{S_i} \mathbf{d}^T d\xi d\eta = \left\{ C_{00} \quad C_{10} \quad C_{01} \quad \frac{1}{2}C_{20} \quad C_{11} \quad \frac{1}{2}C_{02} \right\} \quad (7.27)$$

Determining the integral of the pressure coefficient over the panel now simply reduces to calculating the C integrals. The calculation of these integrals (adapted from [33]) is presented in Appendix I. Once these have been calculated, the total force vector for panel i is found to be

$$\begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix}_i = -J\mathbf{n}_i^g \left\{ C_{00} \quad C_{10} \quad C_{01} \quad \frac{1}{2}C_{20} \quad C_{11} \quad \frac{1}{2}C_{02} \right\} [S_\mu]^{-1} \{C_P\} \quad (7.28)$$

The implementation of these formulas was checked against a symbolic solver, and average pressure coefficient values were found to match out to precision.

7.3.2 Moment Integration

The moment coefficient about the global origin due to the pressure distribution on a panel i is given by

$$C_{M_i} = -\frac{1}{l_{\text{ref}}} \iint_{S_i} \mathbf{Q} \times \mathbf{n}_i C_P dS_i \quad (7.29)$$

where l_{ref} is a reference length, \mathbf{Q} is the integration point on the panel, and \mathbf{n}_i is the panel normal vector. Since all panels are flat, \mathbf{n}_i may be pulled out of the integral, yielding

$$C_{M_i} = \frac{1}{l_{\text{ref}}} \mathbf{n}_i \times \iint_{S_i} \mathbf{Q} C_P dS_i \quad (7.30)$$

Let $\mathbf{P}_{\text{cent}_i}$ be the centroid of panel i . The moment coefficient may then be written as

$$C_{M_i} = \frac{1}{l_{\text{ref}}} \mathbf{n}_i \times \iint_{S_i} (\mathbf{Q} - \mathbf{P}_{\text{cent}_i} + \mathbf{P}_{\text{cent}_i}) C_P dS_i$$

$$C_{M_i} = \frac{1}{l_{\text{ref}}} \left[\mathbf{n}_i \times \iint_{S_i} (\mathbf{Q} - \mathbf{P}_{\text{cent}_i}) C_P dS_i + \mathbf{n}_i \times \iint_{S_i} \mathbf{P}_{\text{cent}_i} C_P dS_i \right]$$

$$C_{M_i} = \frac{1}{l_{\text{ref}}} \left[\mathbf{n}_i \times \iint_{S_i} (\mathbf{Q} - \mathbf{P}_{\text{cent}_i}) C_P dS_i + \mathbf{n}_i \times \mathbf{P}_{\text{cent}_i} \iint_{S_i} C_P dS_i \right]$$

The second integral in this equation may be written in terms of the force coefficient vector, giving

$$C_{M_i} = \frac{1}{l_{\text{ref}}} \left[\mathbf{n}_i \times \iint_{S_i} (\mathbf{Q} - \mathbf{P}_{\text{cent}_i}) C_P dS_i - \begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix} \times \mathbf{P}_{\text{cent}_i} \right]$$

The first integral and \mathbf{Q} may be expressed in the local-scaled coordinate system, yielding

$$C_{M_i} = \frac{1}{l_{\text{ref}}} \left[J_i \mathbf{n}_i \times \iint_{S_i^{ls}} [A_{ls \rightarrow g}] \mathbf{Q}^{ls} C_P d\xi d\eta - \begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix} \times \mathbf{P}_{\text{cent}_i} \right]$$

$$C_{M_i} = \frac{1}{l_{\text{ref}}} \left[J_i \mathbf{n}_i \times [A_{ls \rightarrow g}] \iint_{S_i^{ls}} \mathbf{Q}^{ls} C_P d\xi d\eta - \begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix} \times \mathbf{P}_{\text{cent}_i} \right]$$

Recognizing that $\mathbf{Q}^{ls} = [\xi \ \eta \ 0]^T$ and (from before) $C_P = \mathbf{d}^T [S_\mu]^{-1} \{C_P\}$, this may be written

$$C_{M_i} = \frac{1}{l_{\text{ref}}} \left[J_i \mathbf{n}_i \times [A_{ls \rightarrow g}] \iint_{S_i^{ls}} \begin{Bmatrix} \xi \\ \eta \\ 0 \end{Bmatrix} \mathbf{d}^T [S_\mu]^{-1} \{C_P\} d\xi d\eta - \begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix} \times \mathbf{P}_{\text{cent}_i} \right]$$

The outer product may be expanded to

$$\begin{Bmatrix} \xi \\ \eta \\ 0 \end{Bmatrix} \mathbf{d}^T = \begin{bmatrix} \xi & \xi^2 & \xi\eta & \frac{1}{2}\xi^3 & \xi^2\eta & \frac{1}{2}\xi\eta^2 \\ \eta & \xi\eta & \eta^2 & \frac{1}{2}\xi^2\eta & \xi\eta^2 & \frac{1}{2}\eta^3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.31)$$

Since $[S_\mu]^{-1}\{C_P\}$ is constant with respect to the integration, the first term may be expressed in terms of the C integrals defined previously, as in

$$\begin{aligned} C_{M_i} = & \frac{J_i}{l_{\text{ref}}} \mathbf{n}_i \times [A_{ls \rightarrow g}] \begin{bmatrix} C_{10} & C_{20} & C_{11} & \frac{1}{2}C_{30} & C_{21} & \frac{1}{2}C_{12} \\ C_{01} & C_{11} & C_{02} & \frac{1}{2}C_{21} & C_{12} & \frac{1}{2}C_{03} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} [S_\mu]^{-1}\{C_P\} \\ & + \frac{1}{l_{\text{ref}}} \mathbf{P}_{\text{cent}_i} \times \begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix} \end{aligned} \quad (7.32)$$

Again, the calculation of the necessary C integrals is discussed in Appendix I.

Equation (7.32) may be applied to both quadratic-doublet-linear-source and linear-doublet-constant-source panels, though it may be simplified for the latter case. As discussed previously, a linear-doublet-constant-source panel will have a constant pressure distribution on its surface. This means

$$[S_\mu]^{-1}\{C_P\} = \begin{Bmatrix} C_{P_i} \\ 0 \\ 0 \end{Bmatrix} \quad (7.33)$$

The matrix of C integrals in Eq. (7.32) will also only have three columns since the quadratic terms for the pressure distribution are identically zero. Applying Eq. (7.33) to Eq. (7.32) results in

$$C_{M_i} = \frac{J_i C_{P_i}}{l_{\text{ref}}} \mathbf{n}_i \times [A_{ls \rightarrow g}] \begin{Bmatrix} C_{10} \\ C_{01} \\ 0 \end{Bmatrix} + \frac{1}{l_{\text{ref}}} \mathbf{P}_{\text{cent}_i} \times \begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix} \quad (7.34)$$

However, since the origin of the local-scaled coordinate system is the area centroid of the panel, by definition

$$C_{10} = C_{01} = 0 \quad (7.35)$$

Thus, the moment coefficient for a linear-doublet-constant- source panel is simply

$$C_{M_i} = \frac{1}{l_{\text{ref}}} \mathbf{P}_{\text{cent}_i} \times \begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix} \quad (7.36)$$

7.4 Accuracy Determination Based on Pressure Rules

As stated in the introduction, for compressible flows, the Prandtl-Glauert equation represents a linearized approximation to the full potential equation (Eq. (1.48)). The full potential equation governs the flow of an isentropic, perfect fluid that begins in a uniform flow. The Prandtl-Glauert equation is derived from the full potential equation by assuming small perturbations from the uniform onset flow. In the limit of infinitesimally small perturbations, the Prandtl-Glauert equation and the full potential equation should match exactly.

Since the Prandtl-Glauert equation is an approximation to a nonlinear PDE, it would be valuable to know how close the solutions obtained from the Prandtl-Glauert equation are to the “true” solution to the full potential equation (keeping in mind that no fluid is perfect, no flow is truly isentropic, and no onset flow is perfectly uniform). It would be especially useful to be able to do this using only the solution obtained from the Prandtl-Glauert equation.

An approximate measure of the accuracy of the Prandtl-Glauert solution may be obtained heuristically from the various pressure rules listed previously. Erickson [50] states that, “It is good practice to always compute the pressures by [the isentropic and second-order pressure] rules; a substantial disagreement in the results is a sure sign that the small-perturbation assumptions of the Prandtl-Glauert equation are being violated.” This makes good sense. The isentropic pressure rule makes no more assumptions about the flow than the full potential equation. And so if this rule agrees closely with the more approximate pressure rules, then it may be assumed that the Prandtl-Glauert equation is modeling a good approximation to the true nonlinear flow. This rule of thumb is often employed by users of panel methods [92].

CHAPTER 8

NUMERICAL BEHAVIORS AND SENSITIVITIES

To test the sensitivity of the [method], the supersonic flow over a [delta wing] was computed using the arbitrary, somewhat pathological, paneling shown in figure 54a. - Ehlers et al. [2]

Panel methods do not solve the BIE (Eq. (1.125)) exactly. Rather, they solve a discrete approximation (Eq. (1.128)). Hence, it is necessary to examine the effect of the discretization on the solutions obtained.

With all aspects of the numerical behavior, it will be examined how each is affected by the singularity distributions (lower- or higher-order) and boundary condition formulation (Morino or source-free) used. These are labeled in each study as given in Table 8.1. Particular attention is paid here to supersonic cases, but subsonic cases will also be presented in order to give a full picture of the performance of the method as described.

Table 8.1: Labels used for the different combinations of boundary condition formulation and singularity distribution used in this work.

Label	Boundary Condition	Distribution
MH	Morino	higher-order
ML	Morino	lower-order
SH	source-free	higher-order
SL	source-free	lower-order

8.1 Sensitivity to Control Point Offset

One important discretization not explicitly expressed in Eq. (1.128) is the fact that the inner potential conditions are enforced only at discrete points within the configuration, called control points. As discussed in Chapter 3, the control points are placed within the configuration, rather than on the configuration surface, to avoid having to calculate the mesh solid angles at each vertex. The locations of these control points are dependent upon the control point offset (k_1 in Eq. (3.15)). The control point offset is a free parameter, and so its effect on solutions must be examined. The effect of control point offset is considered here first as it must be set in order to determine all subsequent sensitivities.

Maruyama et al. (the developers of MARCAP) point out that, as long as the control points are inside the configuration, there are few mathematical restrictions on their placement [66]. However, Davis showed that computed results can be sensitive to how close the control points are to the configuration surface. Originally, k_1 was implemented in MachLine as a nondimensional parameter, and the actual control point offset was the product of this ratio and the average of the edge lengths touching the vertex (see Eq. (3.16)). This meant that in regions of a mesh where the panels were smaller, the control points would be closer to the surface of the mesh. The approach was first taken by Davis with the code CPanel. Using this approach, Davis recommended a value of $k_1 = 10^{-4}$ [7].

As will be shown here, specifying k_1 as a ratio makes the appropriate values of k_1 dependent on the mesh dimensions and resolution. Instead, the control point offset should be specified directly without considering edge lengths (as in Eq. (3.15)). Doing so makes the control points equally-distant from the mesh surface, regardless of panel dimensions.

8.1.1 Incompressible Sphere

For a sphere in inviscid, incompressible flow, the total force acting on the sphere should be zero. Thus the effect of control point offset may be analyzed using the norm of the force coefficient vector. Fig. 8.1 shows the force coefficient vector magnitude as a function of control point offset ratio (k_1) varying from 10^{-12} to 1. There was clearly a range of k_1 where

the results did not change with k_1 . However, this range varied somewhat between the two mesh densities, showing that specifying k_1 as a ratio introduced some mesh dependence.

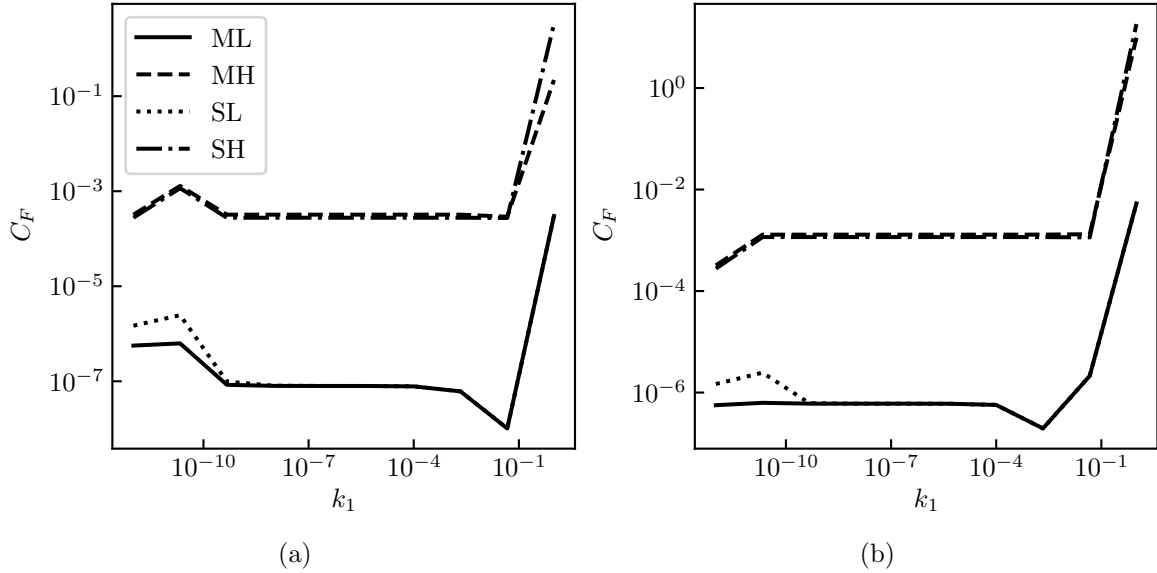


Fig. 8.1: Norm of force coefficient vector as a function of control point offset (specified as a ratio of the average edge lengths) for a sphere with (a) a mesh with approximately 7200 panels and (b) a mesh with approximately 1700 panels.

Alternatively, the control point sensitivities with k_1 specified directly (as in Eq. (3.15)) are shown in Fig. 8.2. In this case, the results were much more uniform between the two mesh densities. In the range of $10^{-10} \leq k_1 \leq 10^{-4}$, there was minimal variation in the force coefficients. This range was also the same between all boundary condition formulations and singularity distributions.

8.1.2 Supersonic Double-Wedge Wing

The sensitivity to control point offset was also examined for a double-wedge wing in supersonic flow. Unlike the sphere in incompressible flow, a wing in supersonic flow will have a nonzero drag force even at zero degrees angle of attack. As such, the sensitivities of the axial and off-axis forces are examined here separately.

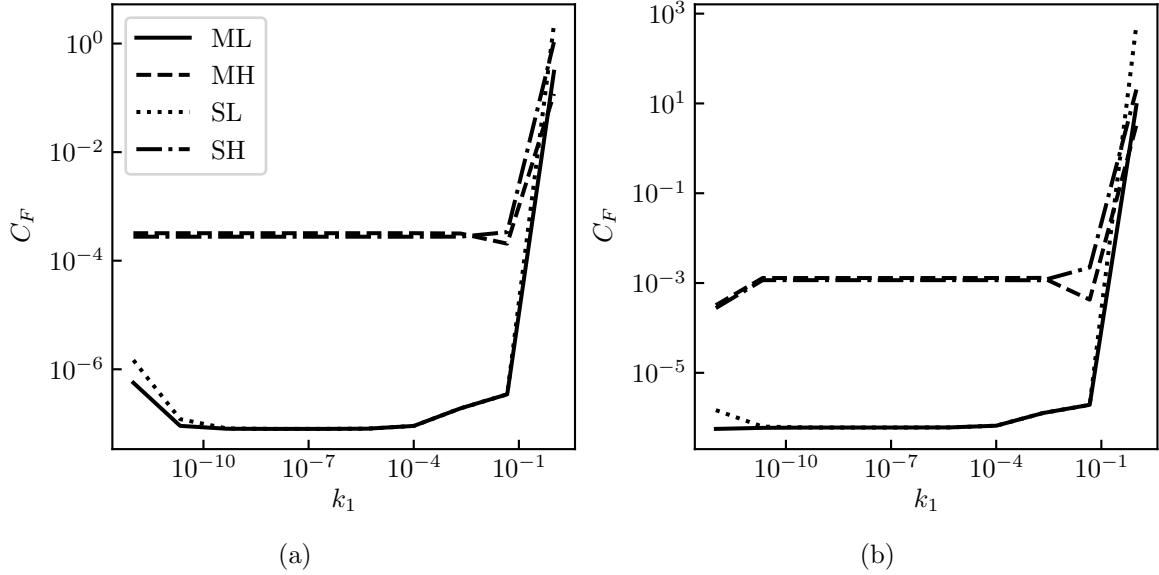


Fig. 8.2: Norm of force coefficient vector as a function of control point offset (specified directly) for a sphere with (a) a mesh with approximately 7200 panels and (b) a mesh with approximately 1700 panels.

The control point offset sensitivities for the double-wedge wing in supersonic flow with k_1 specified as a ratio are shown in Fig. 8.3. These results were obtained with a finely-paneled mesh with approximately 2200 panels and an average characteristic length of 5.6×10^{-2} (the characteristic length here is the square root of the panel area). Over the range of $10^{-8} \leq k_1 \leq 10^{-4}$, the off-axis forces were minimal, and the axial force showed minimal variation for all boundary condition formulations and singularity distributions.

Figure 8.4 shows the sensitivities to control point offset with k_1 specified directly on the same mesh as before. For this particular mesh, there was little qualitative change compared to when k_1 was specified as a ratio (Fig. 8.3). However, there was a shift in the region of minimal sensitivity. For the direct specification, the least control point sensitivity was seen between $10^{-10} \leq k_1 \leq 10^{-5}$. The magnitude of this shift makes sense considering the average characteristic length for the mesh.

To examine the sensitivity of this range to the mesh, the same study was run for a coarser version of the double-wedge wing mesh (approximately 800 panels and an average characteristic length of 9×10^{-2}). The results are shown in Fig. 8.5. In this case, the

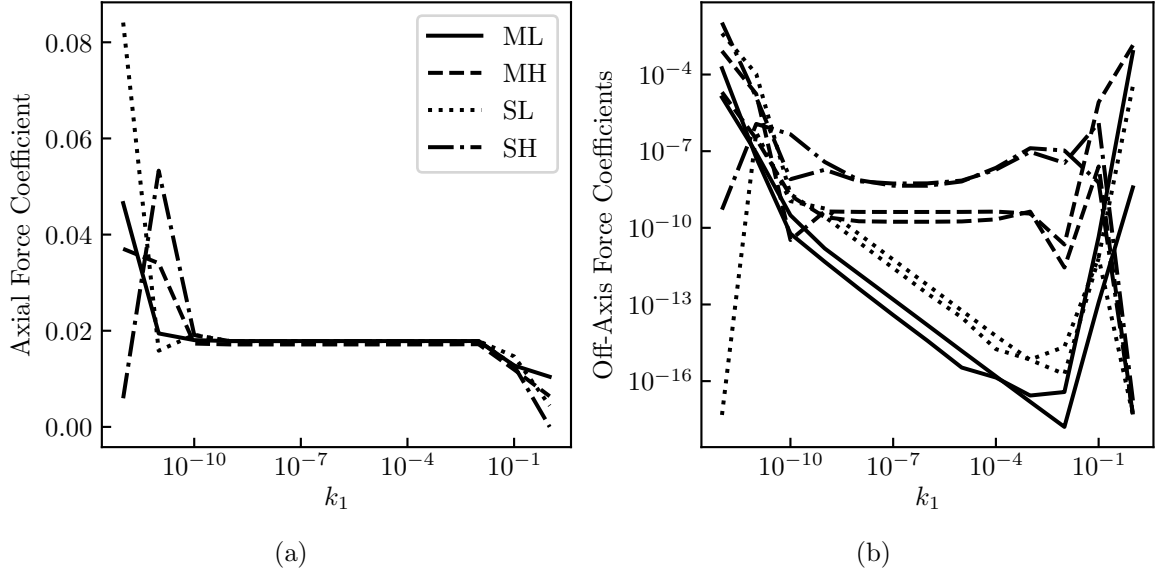


Fig. 8.3: (a) Axial and (b) off-axis force coefficient magnitudes on a double-wedge wing as a function of k_1 (specified as a ratio) for all boundary condition formulations and singularity distributions.

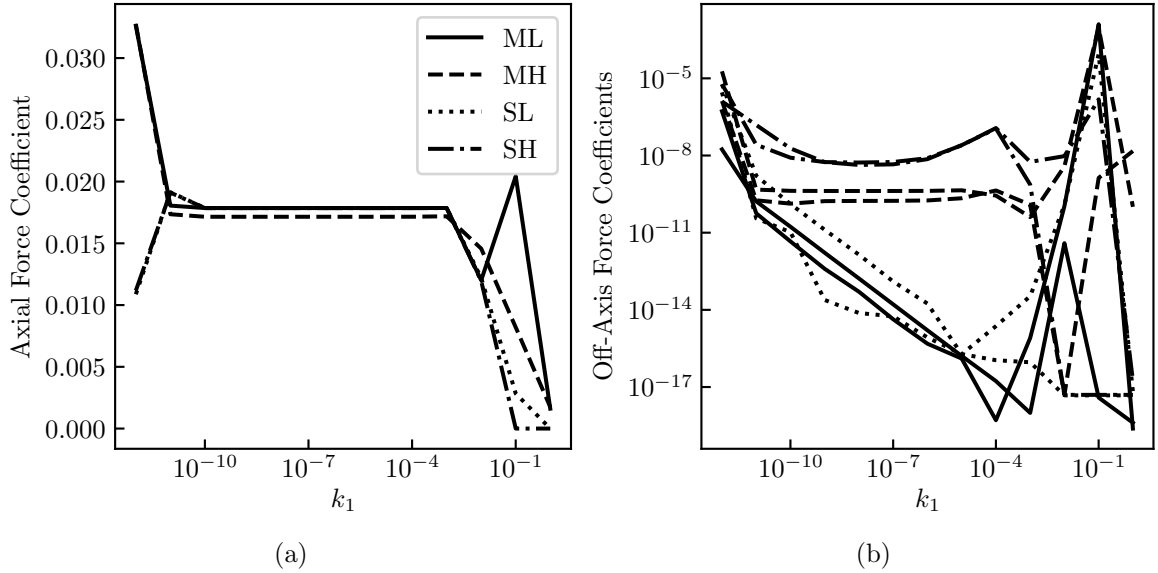


Fig. 8.4: (a) Axial and (b) off-axis force coefficient magnitudes on a finely-meshed (2200 panels) double-wedge wing as a function of k_1 (specified directly).

qualitative behavior was still the same as both previous cases. In addition, the range of minimal sensitivity was practically the same as that for the more-refined mesh. This again

indicated that specifying the control point offset directly, instead of as a ratio of average lengths, is more appropriate.

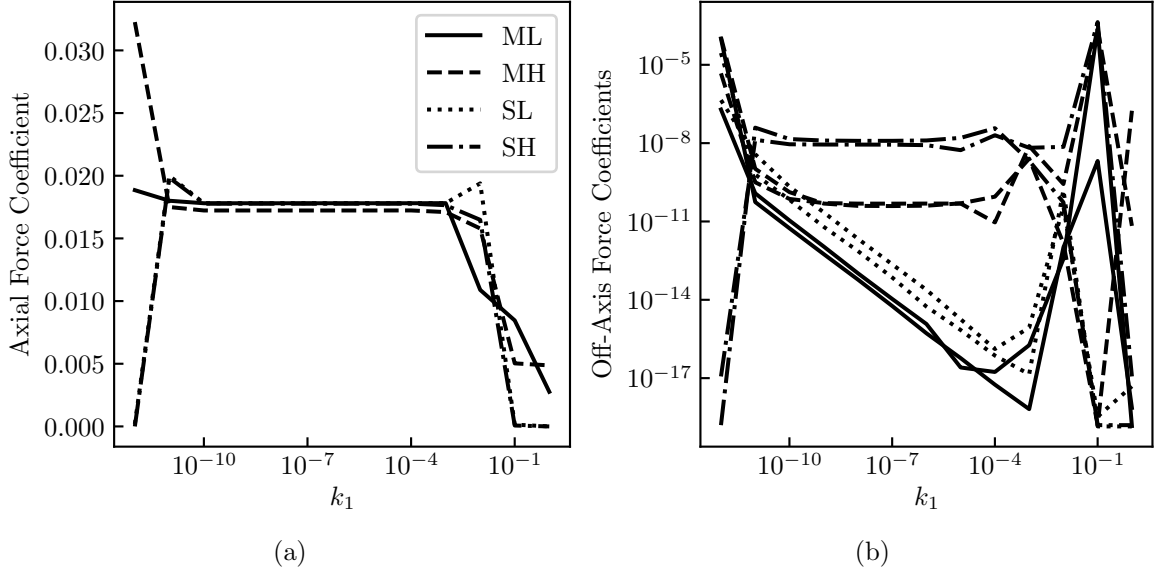


Fig. 8.5: (a) Axial and (b) off-axis force coefficient magnitudes on a coarsely-meshed (800 panels) double-wedge wing as a function of k_1 , with k_1 specified as the offset directly, rather than as a ratio.

8.1.3 Supersonic Spindle

To verify the results obtained for the incompressible sphere and the supersonic double wedge wing, the sensitivities to control point offset were also examined for a parabolic body of revolution (called a *spindle*) in supersonic flow. This spindle had a unit chord length and a radial thickness distribution given by

$$r(x) = \frac{1}{5}x(1-x) \quad (8.1)$$

with the x -axis being the axis of revolution.

As with the double-wedge wing, a spindle in supersonic flow at zero angle of attack will still experience a finite drag force. Because of this, the axial and off-axis forces were

examined separately. For this configuration, only the case where k_1 specifies the control point offset directly was considered.

Figure 8.6 shows these results for a fine mesh (12500 panels), and Fig. 8.7 shows these results for a coarse mesh (722 panels). In both cases, the results were minimally sensitive to control point offset in the range $10^{-10} \leq k_1 \leq 10^{-4}$, as with the incompressible sphere and the supersonic double-wedge wings.

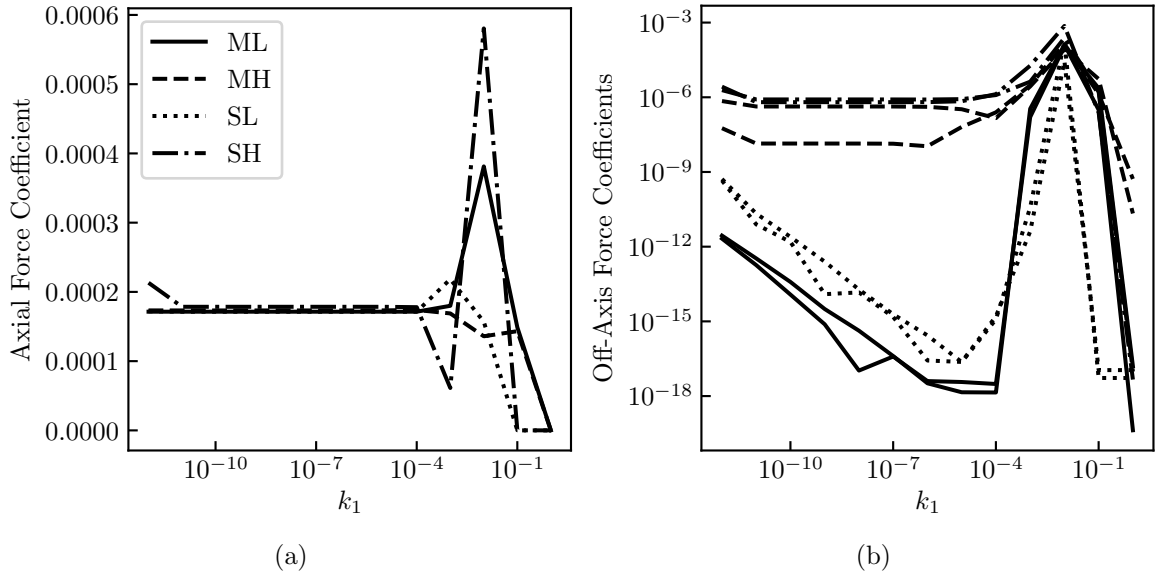


Fig. 8.6: (a) Axial and (b) off-axis force coefficient magnitudes on a finely-meshed (12500 panels) supersonic spindle as a function of k_1 .

8.1.4 Discussion

Between the cases considered here, setting k_1 in the range of 10^{-10} to 10^{-4} resulted in minimal sensitivity to k_1 . This range was the same for all boundary condition formulations and singularity distributions. In addition, it was determined that specifying the control point offset directly, rather than as a ratio to average edge lengths, is more reliably mesh-independent. Because of this, the default control point offset in MachLine is 10^{-7} , specified directly. For all subsequent cases, $k_1 = 10^{-7}$, unless otherwise noted.

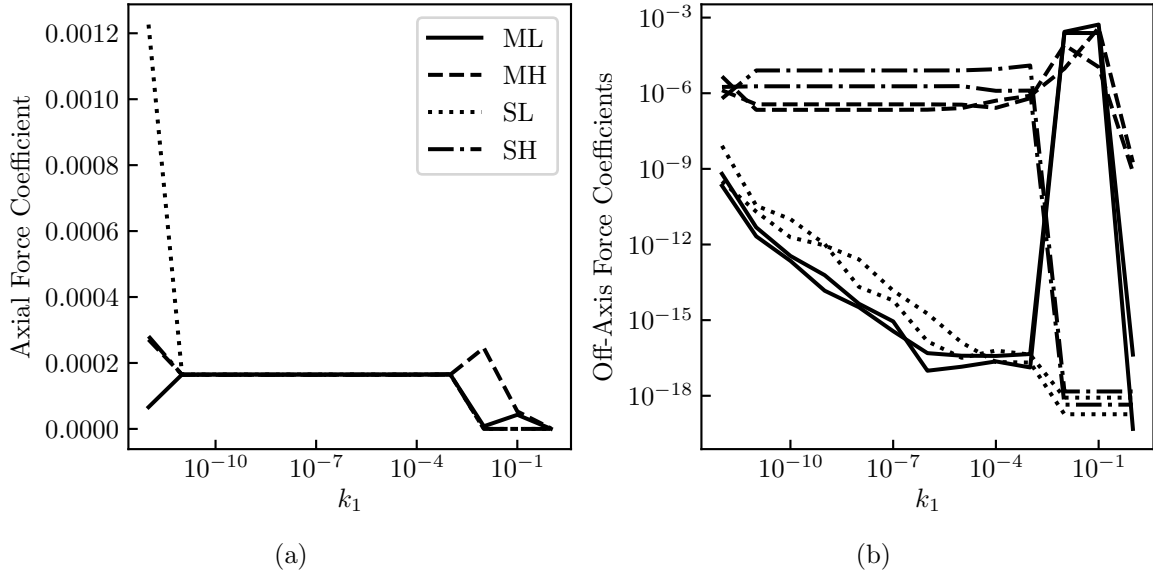


Fig. 8.7: (a) Axial and (b) off-axis force coefficient magnitudes on a coarsely-meshed (722 panels) supersonic spindle as a function of k_1 .

There was some slight variation in the range for which the control point offset gave minimal sensitivity. Thus, it is conceivable that this range is different for highly unusual configurations. If highly accurate results are required, then it is recommended that the effect of control point offset be examined for individual cases. However, the results presented here may serve as a baseline.

8.2 Order of Grid Convergence

After the default control point offset has been established, the most important numerical aspect to consider is that of grid convergence. As the discretization in Eq. (1.128) is refined (i.e. the panels are made smaller), the solution should everywhere converge to some fixed distribution. If this is not the case, then the problem has been formulated incorrectly.

If the solution does converge, then an estimate may be made as to the order of convergence. The order of convergence indicates the rate at which the error in the solution decreases as the computational mesh is refined. The measure of mesh refinement here is the average characteristic length of all the panels in the mesh, the characteristic length being

the square root of the panel area. A coarse mesh will have a larger average characteristic length than a fine mesh.

The order of convergence is determined by fitting a power law to the error in the solution against the average mesh characteristic length. Exactly what the error is depends on the situation, but it is often the difference between the calculated result and some known analytic solution or a result obtained using a finer computational mesh. The exponent of the resulting power fit is then the estimate of the order of convergence.

8.2.1 Incompressible Sphere

The first case used here to determine the order of convergence of MachLine for the different boundary condition formulations and singularity distributions was a sphere in incompressible flow. Since the total force on a sphere in incompressible flow should be zero, the norm of the total force vector acting on the sphere is a direct measure of the numerical error. To analyze the order of grid convergence for a sphere, four different levels of mesh refinement were considered. These meshes were all radially symmetric about the x -axis, as shown in Fig. 8.8. Because of this, the total force vector was very close to machine zero regardless of the mesh refinement if the freestream velocity was aligned with the x -axis. Thus, to properly analyze grid convergence, the sphere was tested at three different (non-zero) angles of attack and sideslip angles, each, resulting in nine different freestream orientations for each level of mesh refinement.

The norm of the force coefficient vector acting on the sphere as a function of the average panel characteristic length is shown in Fig. 8.9. Interestingly, over the nine freestream orientations considered, there were no variations in force coefficients for a given singularity distribution and boundary condition formulation. Hence the appearance of only one line for each case in Fig. 8.9. The total force magnitude was much higher over all mesh densities when the higher-order singularity distributions were used compared to the lower-order singularity distributions. In addition, the lower-order singularity distributions resulted in faster grid convergence than the higher-order singularity distributions. The source-free formulation resulted in somewhat faster convergence and lower error than the Morino for-

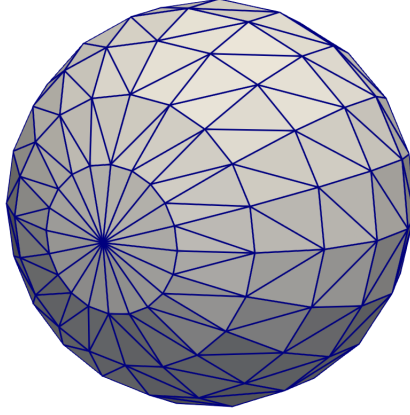


Fig. 8.8: Example sphere mesh showing the radial symmetry about the x -axis.

mulation when higher-order singularity distributions were used. When the low-order singularity distributions were used, the results were the same regardless of boundary condition formulation.

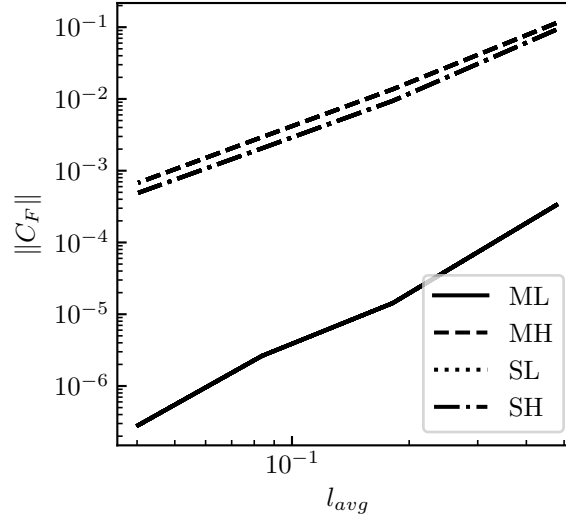


Fig. 8.9: Norm of the force coefficient acting on the sphere as a function of the average panel characteristic length for each singularity distribution and boundary condition formulation.

The orders of convergence for the sphere are given in Table 8.2. As was shown in Fig. 8.9, when using lower-order distributions, both boundary condition formulations resulted in the same order of convergence. Again, the orders of convergence resulting from the

higher-order singularity distributions were significantly lower than those resulting from the lower-order singularity distributions.

Table 8.2: Orders of convergence for the sphere in incompressible flow. The uncertainty band represents one standard deviation.

BC	Singularities	Order of Convergence
Morino	Higher	2.068
Morino	Lower	2.803
Source-free	Higher	2.110
Source-free	Lower	2.803

8.2.2 Supersonic Spindle

The second case used here to examine orders of convergence was a spindle in supersonic flow. This spindle was the same as that considered in Section 8.1.3. The spindle was tested at various Mach numbers and zero angle of attack. In each case, the drag force was used to determine grid convergence. The off-axis forces were not considered as these were close to machine zero regardless of the mesh density. Four different mesh refinement levels were used, with the error being measured relative to the drag on the most-refined mesh.

The relative error in the axial force coefficient as a function of the mesh average characteristic length at various freestream Mach numbers is shown in Fig. 8.10. The grid convergence varied significantly over the range of Mach numbers considered when the higher-order distributions were used. In general, the error was also higher when higher-order distributions were used compared to lower-order distributions. When lower-order distributions were used, the relative error in the axial force coefficient again did not vary between the two boundary condition formulations.

The orders of convergence for the spindle in supersonic flow are given in Table 8.3. As was seen with the sphere in incompressible flow, the convergence rates when using lower-order distributions were identical. And as before, the convergence rates were significantly lowered by the use of higher-order distributions. Unlike with the sphere, the Morino formu-

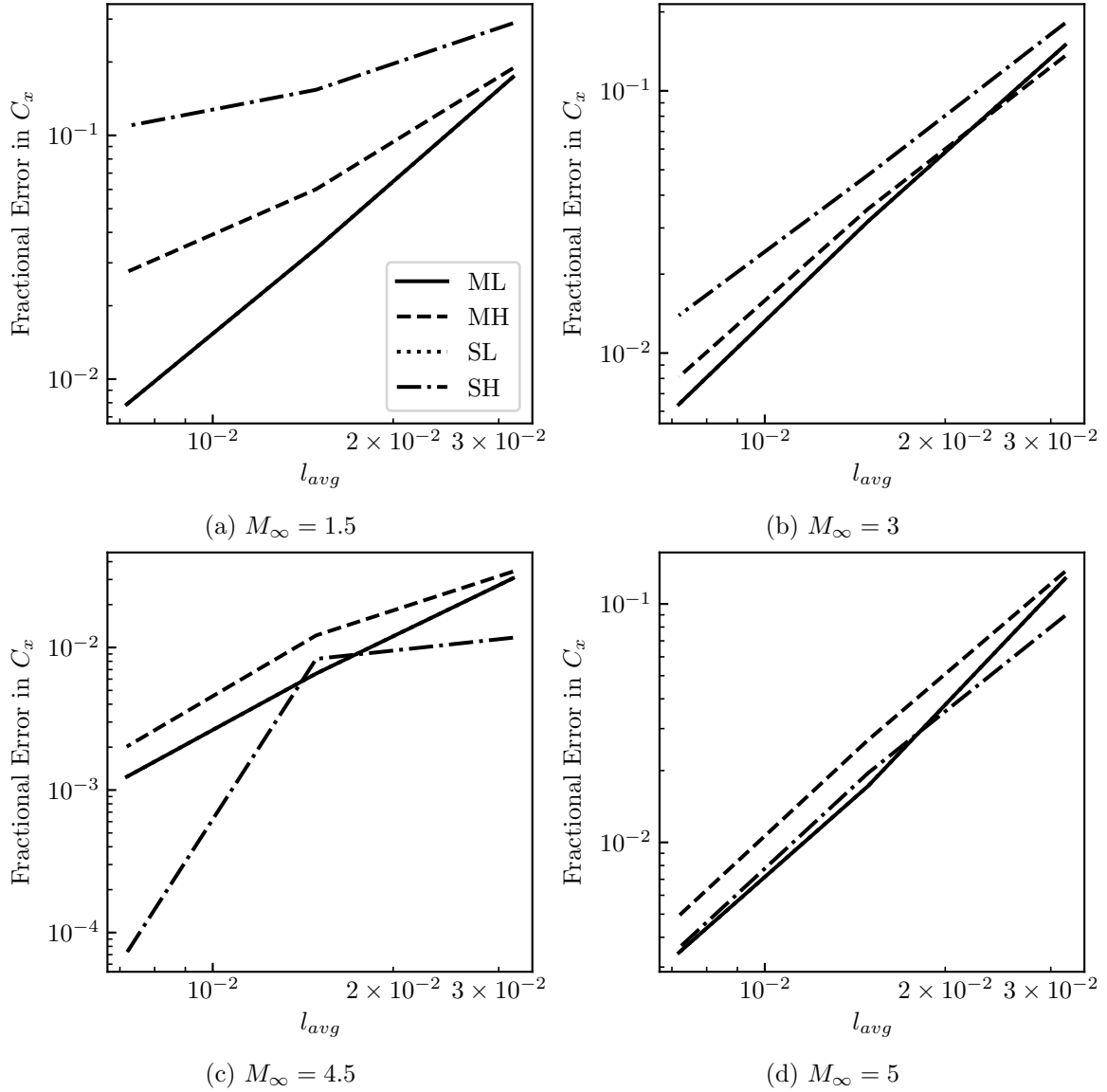


Fig. 8.10: Fractional error relative to the most-refined mesh in the axial force coefficient on the spindle as a function of mesh average characteristic length at various Mach numbers for each of the boundary condition formulations and singularity distributions.

lation paired with higher-order distributions showed faster grid convergence than did the source-free formulation paired with higher-order distributions, though only slightly.

To obtain these results, the maximum continuity angle was 45° . This was deemed appropriate because the spindle was a smooth body that, while never flat, had no sharp breaks in geometry. Thus, having continuous surface velocity and pressure distributions

Table 8.3: Orders of convergence for the spindle in supersonic flow. The uncertainty band represents the standard deviation over all Mach numbers.

BC	Singularities	Order of Convergence
Morino	Higher	1.850 ± 0.279
Morino	Lower	2.19 ± 0.115
Source-free	Higher	1.705 ± 0.77
Source-free	Lower	2.19 ± 0.115

was not deemed an issue. The effect of the maximum continuity angle is examined in the next section.

8.2.3 Supersonic Double-Wedge Wing

The third case used here for examining the order of convergence of MachLine was a double-wedge wing in supersonic flow. This wing had a wedge half angle of 5° , rounded tips, and a large aspect ratio. An example of one of the meshes used for this wing is shown in Fig. 8.11. This wing was tested at four Mach numbers and five angles of attack. In addition, the effect of the maximum continuity angle (see Section 5.5) on order of convergence was also examined. To do this, two different maximum continuity angles were used: 1° and 45° . With a maximum continuity angle of 45° , all edges on the mesh were considered continuous. With a maximum continuity angle of 1° , only the edges within the flat upper and lower surfaces were considered continuous.

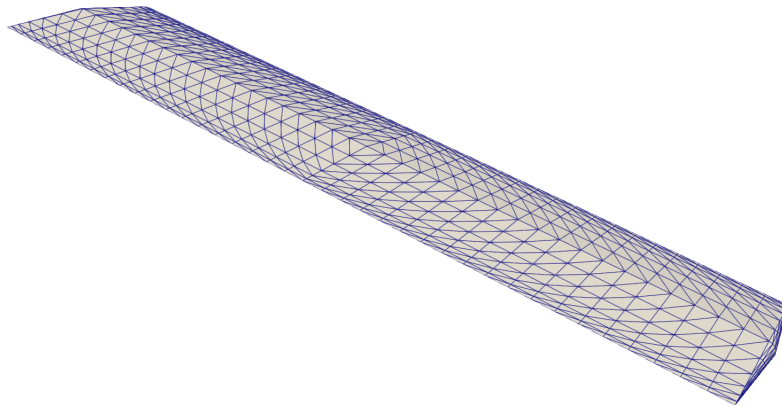


Fig. 8.11: Example mesh for the double-wedge wing.

The force coefficients in the x and z directions, C_x and C_z , were used to determine grid convergence. Four levels of mesh refinement were used, with errors being calculated relative to the most-refined mesh. The orders of convergence were averaged between all Mach numbers and angles of attack. The results for C_z at zero angle of attack were excluded, as C_z was very close to machine zero in this case, regardless of mesh refinement.

The estimated orders of convergence for the double wedge wing are given in Tables 8.4 and 8.5. As with both previous studies, the order of convergence was the same for both boundary condition formulations when lower-order distributions were used. With the lower-order distributions, the order of convergence was also independent of the maximum continuity angle, as would be expected, since the maximum continuity angle does not come into play when lower-order distributions are used. When the lower-order distributions were used, the rate of convergence was higher in C_x than in C_z , though the differences between the two orders were small.

Table 8.4: Orders of convergence for the double-wedge wing in supersonic flow with a maximum continuity angle of 1° . The uncertainty band is the standard deviation across all Mach numbers and angles of attack.

BC	Singularities	Convergence in C_x	Convergence in C_z
Morino	Higher	2.199 ± 0.752	2.301 ± 0.157
Morino	Lower	2.464 ± 0.509	2.199 ± 0.139
Source-free	Higher	2.116 ± 0.427	2.287 ± 0.159
Source-free	Lower	2.463 ± 0.509	2.198 ± 0.139

Table 8.5: Orders of convergence for the double-wedge wing in supersonic flow with a maximum continuity angle of 45° . The uncertainty band is the standard deviation across all Mach numbers and angles of attack.

BC	Singularities	Convergence in C_x	Convergence in C_z
Morino	Higher	1.821 ± 0.924	2.109 ± 0.659
Morino	Lower	2.464 ± 0.509	2.199 ± 0.139
Source-free	Higher	2.739 ± 0.557	1.967 ± 0.582
Source-free	Lower	2.463 ± 0.509	2.198 ± 0.139

For the higher-order singularity distributions, the rates of grid convergence were much better in general when the maximum continuity angle was 1° than when it was 45° . The exception to this was when the source-free formulation was used with higher-order singularity distributions. Combined with a maximum continuity angle of 45° , these resulted in a convergence rate of greater than 2.7 in C_x , the highest rate seen out of any case. However, the convergence rate in C_z was particularly poor for this case. When the Morino formulation was used with the higher-order singularity distributions, the rates of convergence were lower when a maximum continuity angle of 45° was used. In addition, the rates of grid convergence in C_z when the maximum continuity angle was 1° were higher when higher-order distributions were used than when lower-order distributions were used.

This study demonstrates the validity of including the maximum continuity angle. As discussed previously, the use of higher-order singularity distributions results in some level of continuity in surface velocities and pressures between panels. This heightened continuity may at times improve the predicted flow results over flat portions of the mesh where surface pressures and velocities vary but are continuous, such as aft of the tip characteristic line, as shown in Fig. 8.12. Since the mesh is flat, using flat panels is an exact representation of the geometry. However, along the line of maximum thickness, where there is a sharp break in geometry, the surface velocities and pressures should be discontinuous. The inclusion of a maximum continuity angle allowed for this, and the rates of grid convergence were accordingly improved.

An exception to this was that the rate of grid convergence for the source-free formulation paired with higher-order singularity distributions for C_x was higher than any other case considered when the maximum continuity angle was 45° . However, in this case, the convergence rate for C_z was below average. This may indicate that the doublet distributions are less sensitive to the heightened continuity introduced by the higher-order distributions than the source distributions. This might justify examining a formulation where quadratic-doublet-constant-source panels are used with the Morino formulation. However, such an investigation is outside the scope of the current work.

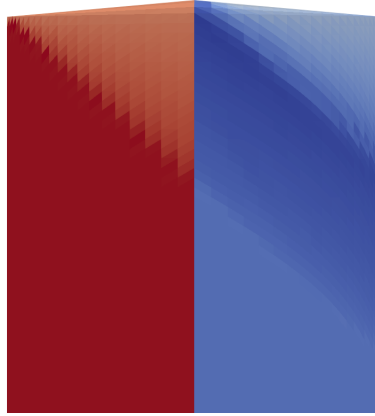


Fig. 8.12: Isentropic pressure distribution at the tip of the double-wedge wing at $M_\infty = 1.5$ and $\alpha = 0^\circ$. The freestream flow is from left to right. Color scale ranges from dark red (high pressure) to dark blue (low pressure).

Some selected convergence charts are shown in Figs. 8.13 and 8.14 for when the maximum continuity angle was 1° . Overall, the levels of error were somewhat higher for the higher-order singularity distributions than for the lower-order singularity distributions. There were not significant differences between the boundary condition formulations at this maximum continuity angle except that, when higher-order distributions were used, the Morino formulation tended to result in less error in C_x than the source-free formulation.

8.2.4 Supersonic Delta Wing

To further examine the grid convergence for configurations with large flat surfaces in supersonic flow, a delta wing with a diamond section was considered. This wing had 45° leading-edge sweep, a maximum thickness of 8% located at 18% chord, and a taper ratio of zero (modeled after wing 11 in [6]). As such, the computational mesh consisted of six exactly flat facets, divided into panels. Five different levels of mesh refinement were considered. The finest and coarsest meshes are shown in Fig. 8.15.

For this study, the maximum continuity angle for the higher-order distributions was set to 1° . Because of this, the edges along the leading and trailing edges, as well as the line of maximum thickness, were marked as discontinuous. All meshes were tested at $M_\infty = 1.62$ and angles of attack ranging from -6° to 5° . As with the double-wedge wing, the orders

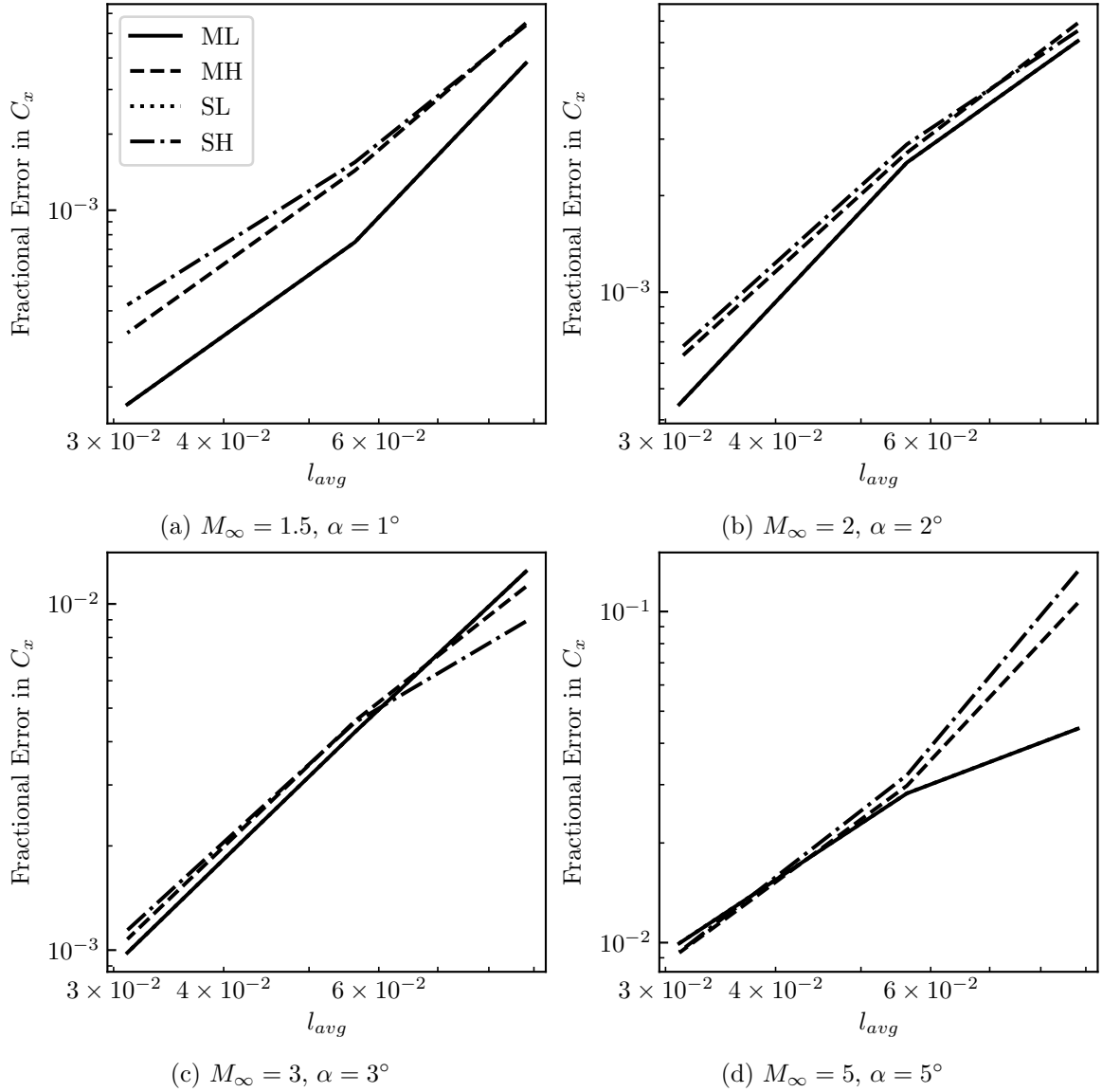


Fig. 8.13: Fractional error relative to the most-refined mesh in the axial force coefficient on the double-wedge wing as a function of mesh average characteristic length at various Mach numbers and angles of attack for each of the boundary condition formulations and singularity distributions.

of convergence were determined based on the error in axial and normal force coefficients relative to the most-refined mesh. In addition, the normal force coefficient at zero angle of attack was not used to determine mesh convergence, as this was always close to machine zero regardless of mesh resolution.

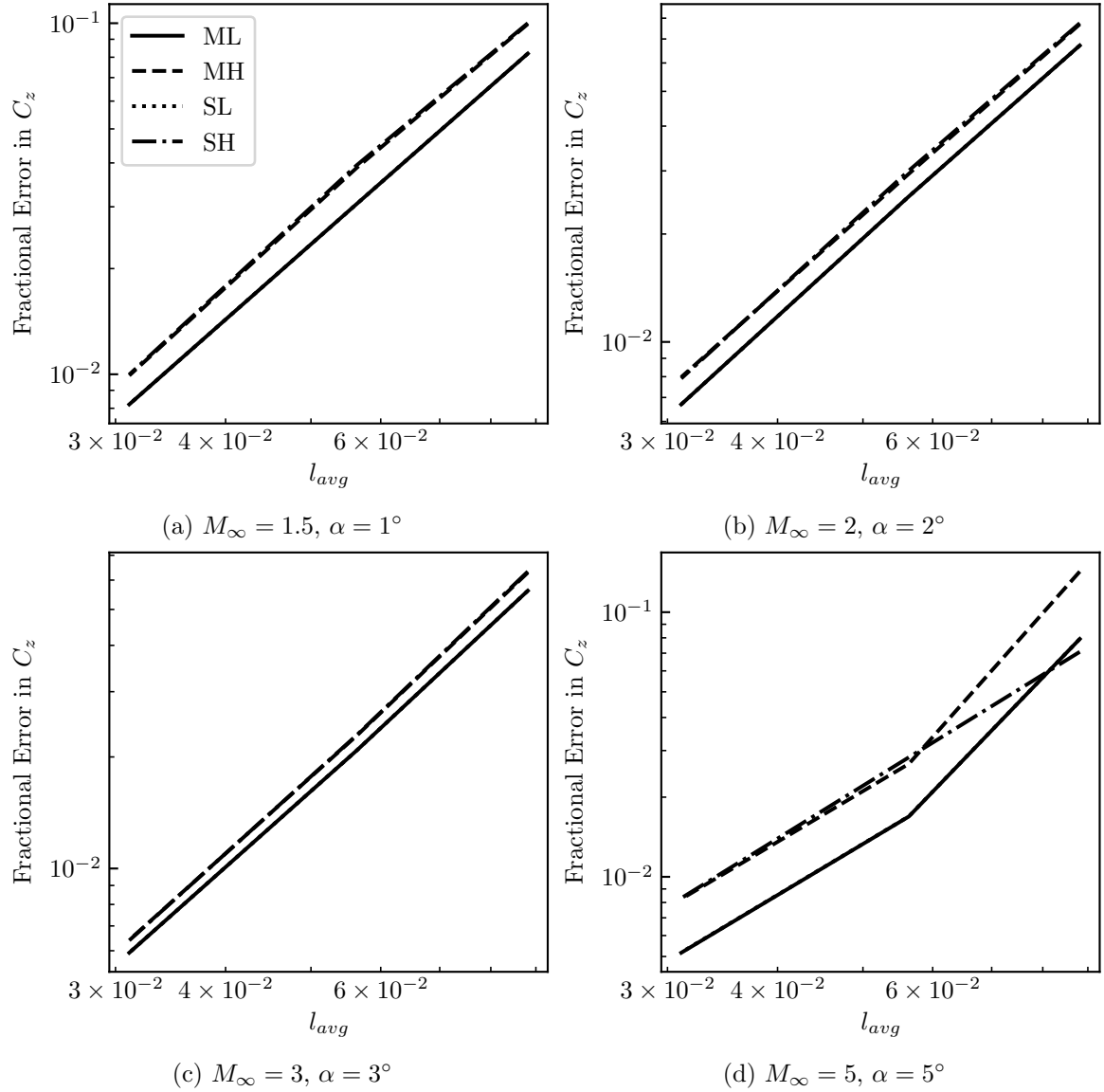


Fig. 8.14: Fractional error relative to the most-refined mesh in the normal force coefficient on the double-wedge wing as a function of mesh average characteristic length at various Mach numbers and angles of attack for each of the boundary condition formulations and singularity distributions.

The grid convergence rates for the delta wing at each angle of attack are shown in Fig. 8.16. The grid convergence in the axial force coefficient, C_x , was very well-behaved for all singularity distributions and boundary condition formulations. Over the range of angles of attack considered, the rate of grid convergence in C_x using higher-order distributions was almost constant, while the rate of grid convergence in C_x varied significantly with

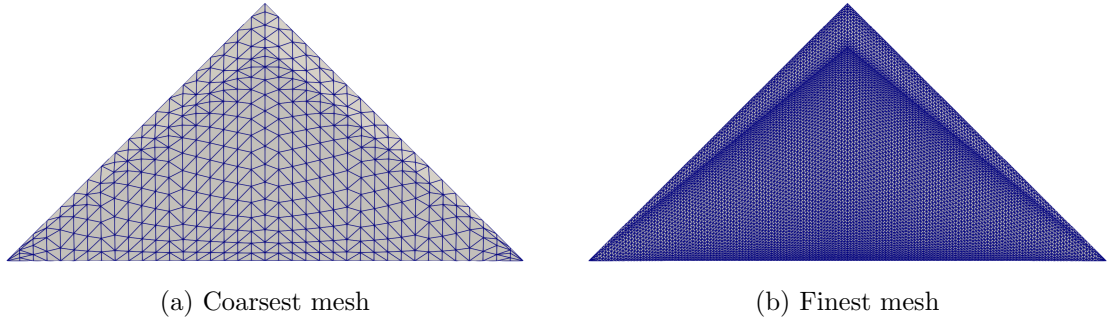


Fig. 8.15: Computational meshes used for the delta wing grid convergence study.

angle of attack when lower-order distributions were used. At smaller angles of attack, the lower-order distributions resulted in higher rates of grid convergence, while the higher-order distributions resulted in higher rates of grid convergence at larger angles of attack. The larger angles of attack would result in stronger shocks, meaning larger gradients in the flow properties across the wing, which could be better captured by the higher-order distributions.

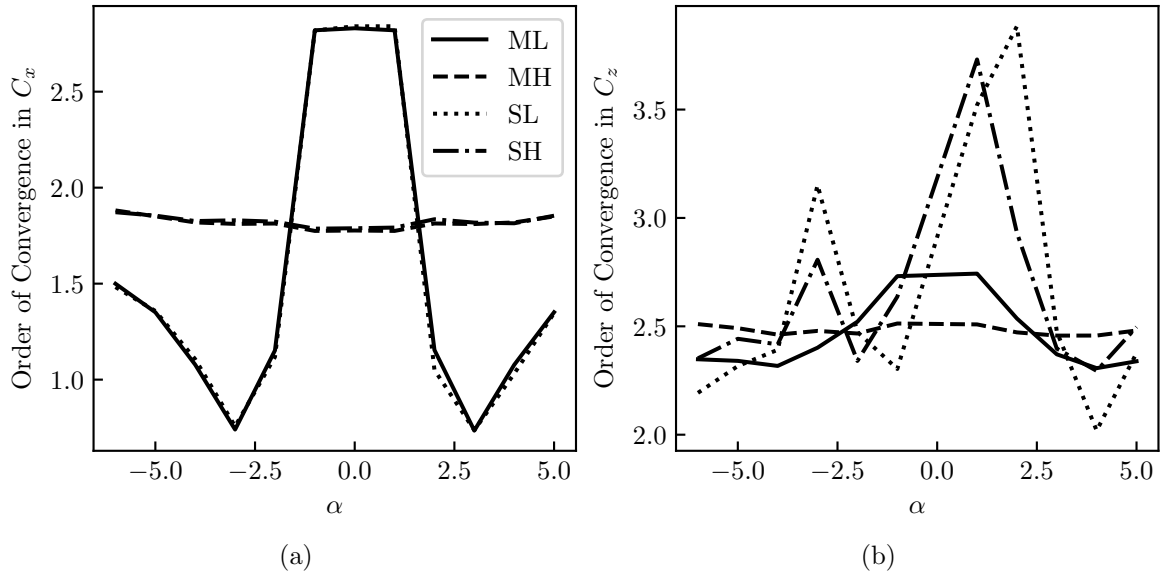


Fig. 8.16: Grid convergences rates for the supersonic delta wing over a range of angles of attack.

For the normal force (C_z), the rates of grid convergence were not as well-behaved. Significant variations in rate of grid convergence with angle of attack were shown for all

cases except when the higher-order distributions were paired with the Morino formulation. Looking at just the Morino formulation, the lower-order distributions resulted in higher rates of grid convergence for smaller angles of attack, and the higher-order distributions resulted in higher rates of grid convergence for larger angles of attack (as was the case with C_x). For the source-free formulation, the rates of grid convergence showed less variation between the two singularity distributions.

The orders of convergence averaged across all angles of attack for the delta wing are listed in Table 8.6. In all cases except for the source-free formulation for C_z , the use of higher-order singularity distributions resulted in a higher average rate of grid convergence than the use of lower-order distributions. For the case of the source-free formulation, the rates of grid convergence in C_z differed between singularity distributions by only 0.024, which was significantly less than the uncertainty of the estimate. A more thorough study could possibly reveal the superiority of the higher-order distributions in this case. Of course, these rates of grid convergence were averaged over a range of angles of attack. Considering different angles of attack or freestream Mach numbers would likely alter the estimated orders of convergence.

Table 8.6: Orders of convergence for the delta wing in supersonic flow. The uncertainty band the standard deviation across all angles of attack.

BC	Singularities	Convergence in C_x	Convergence in C_z
Morino	Higher	1.816 ± 0.0312	2.482 ± 0.02
Morino	Lower	1.551 ± 0.765	2.451 ± 0.154
Source-free	Higher	1.825 ± 0.026	2.622 ± 0.4
Source-free	Lower	1.541 ± 0.776	2.646 ± 0.57

8.2.5 Discussion

Overall, the higher-order distributions showed poorer grid convergence than the lower-order distributions, regardless of the boundary condition formulation. For both the sphere in incompressible flow and the spindle in supersonic flow, the convergence rates were lower with higher-order distributions than with lower-order distributions.

The results in this section confirmed that the use of higher-order singularity distributions is not sufficient for producing a panel method that is truly “higher-order”. As discussed by Hess [52] and in Chapter 5 in this work, a linear variation of source strength and a quadratic variation of doublet strength both have an effect of the same order as panel curvature. In order to have a truly higher-order panel method, panel curvature must also be taken into account. Since, the panel influence coefficients cannot be evaluated analytically for curved panels in supersonic flow [2], the best that can be done with a supersonic panel method are higher-order distributions of source and doublet strength on flat panels.

However, for the double-wedge and delta wings in supersonic flow, the higher-order distributions sometimes resulted in higher rates of grid convergence when the maximum continuity angle was very small. On the double-wedge wing, there were large flat surfaces that also had significant variations in the flow due to tip effects, as shown in Fig. 8.12. For these regions, the fact that the panels were flat did not reduce the accuracy of the discretization, as the true geometry was also flat. For these flat surfaces, the source distribution was constant regardless of whether higher- or lower-order distributions were used (note the similarity in order of convergence between both boundary condition formulations when the maximum continuity angle was 1°). But there was also significant variation in the surface properties, and so the higher-order doublet distributions were better able to capture these variations than the lower-order doublet distributions.

This was again shown for the delta wing, which consisted of only perfectly flat surfaces joined by sharp angles. With the maximum continuity angle set to 1° , the higher-order distributions resulted in higher rates of grid convergence at most angles of attack compared to the lower-order distributions. Even more than with the double-wedge wing, which had

rounded tips, the panel geometry was an exact representation of the delta wing in this case, making the use of higher-order singularity distributions truly “higher-order”. These studies indicate that the higher-order distributions can provide an increase in accuracy compared to lower-order distributions if the enforcement of continuity between panels can be carefully controlled. They show particular promise for modeling flow over configurations with large flat surfaces where the use of flat panels is not an approximation.

These results also indicate a potential for adaptive selection of singularity distribution. For the rounded tips of the double-wedge wing mesh, the lower-order distributions are likely more appropriate. However, the tip interaction regions, where the geometry is truly flat and significant variations in surface properties exist, may be more accurately modeled by higher-order distributions. The singularity distributions have been formulated such that the order of distribution may be mixed between different panels in the mesh. This is a potential area for future research.

8.3 Sensitivity to Panel Regularity

The final aspect of numerical behavior considered here is sensitivity to panel regularity. An unstructured panel method allows the user great flexibility in defining the surface mesh. The arrangement, shape, and size of the panels used in the mesh are almost entirely arbitrary. As such, it is valuable to determine how sensitive the panel method solutions are to the regularity of the mesh.

8.3.1 Incompressible Sphere

The first case used to analyze sensitivity to panel regularity here was a sphere in incompressible flow. To generate the mesh, a Python script was written which would randomly place a specified number of points on the surface of a sphere and then create panels between them using a convex hull algorithm. The resulting mesh was stored as a VTK file and used in MachLine.

For a sphere in inviscid, incompressible flow, the surface pressure coefficient is given by [141]

$$C_P = 1 - \frac{9}{4} \sin^2 \theta \quad (8.2)$$

where

$$\theta = \arccos x \quad (8.3)$$

for a sphere with unit radius and the freestream velocity aligned with the x -axis. In incompressible flow, a panel method should predict a pressure distribution the same as that given by Eq. (8.2). Any errors will be due to the discretization necessarily introduced by the panel method since the governing equation is the same. In this case, the randomness of the discretization is considered in particular.

The results for a relatively coarse mesh (approximately 500 panels) are shown in Fig. 8.17. In general, the pressures predicted by MachLine were close to the analytic result. For both the Morino and source-free formulations, the lower-order distributions did not ever cause a significant over-prediction of pressure coefficient. However, the higher-order distributions resulted in higher pressures than would be expected (see Figs. 8.17a and 8.18c). On the other hand, the Morino formulation paired with higher-order distributions did not ever significantly under-predict the pressure coefficient.

The results for a fine mesh (3995 panels) are shown in Fig. 8.18. As would be expected, the predicted pressures were much closer to the analytic result for the fine mesh than for the coarse mesh with all formulations and boundary conditions. The Morino formulation paired with higher-order distributions (Fig. 8.18a) followed the analytic solution most closely, whereas the source-free formulation paired with higher-order distributions (Fig. 8.18c) showed the most variation from the analytic solution.

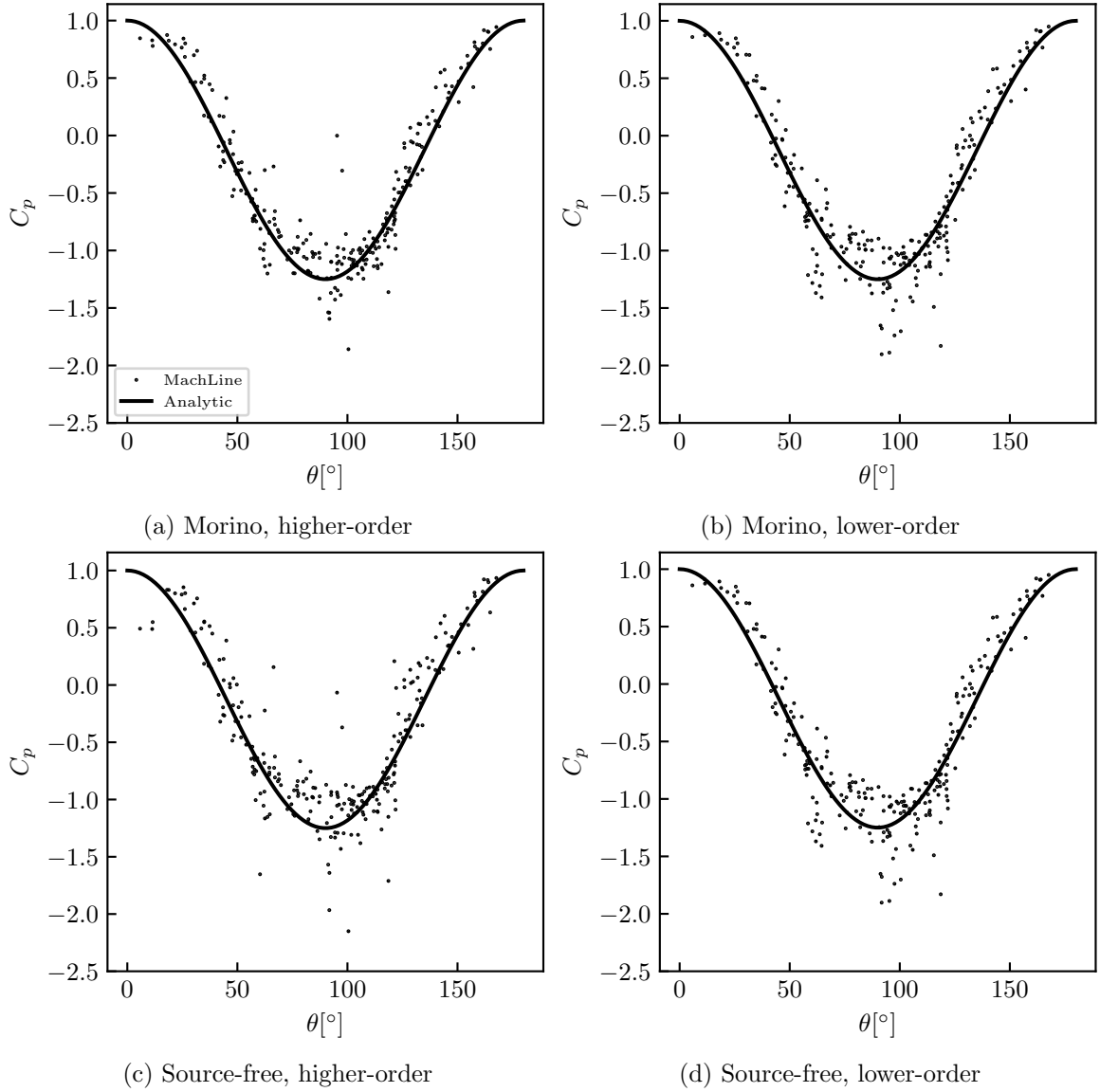


Fig. 8.17: Pressure coefficient predictions for a coarse, randomly-paneled sphere in incompressible flow compared to the analytic result.

8.3.2 Supersonic Spindle

The second case used here to evaluate the sensitivity of MachLine to panel regularity was a spindle in supersonic flow. The spindle considered here had the same radial distribution as that considered previously (see Eq. (8.1) in Section 8.1.3), but the computational mesh used for this study was deliberately randomized. Similar to the randomly paneled sphere, a Python script was written that randomly distributed vertices on the surface of

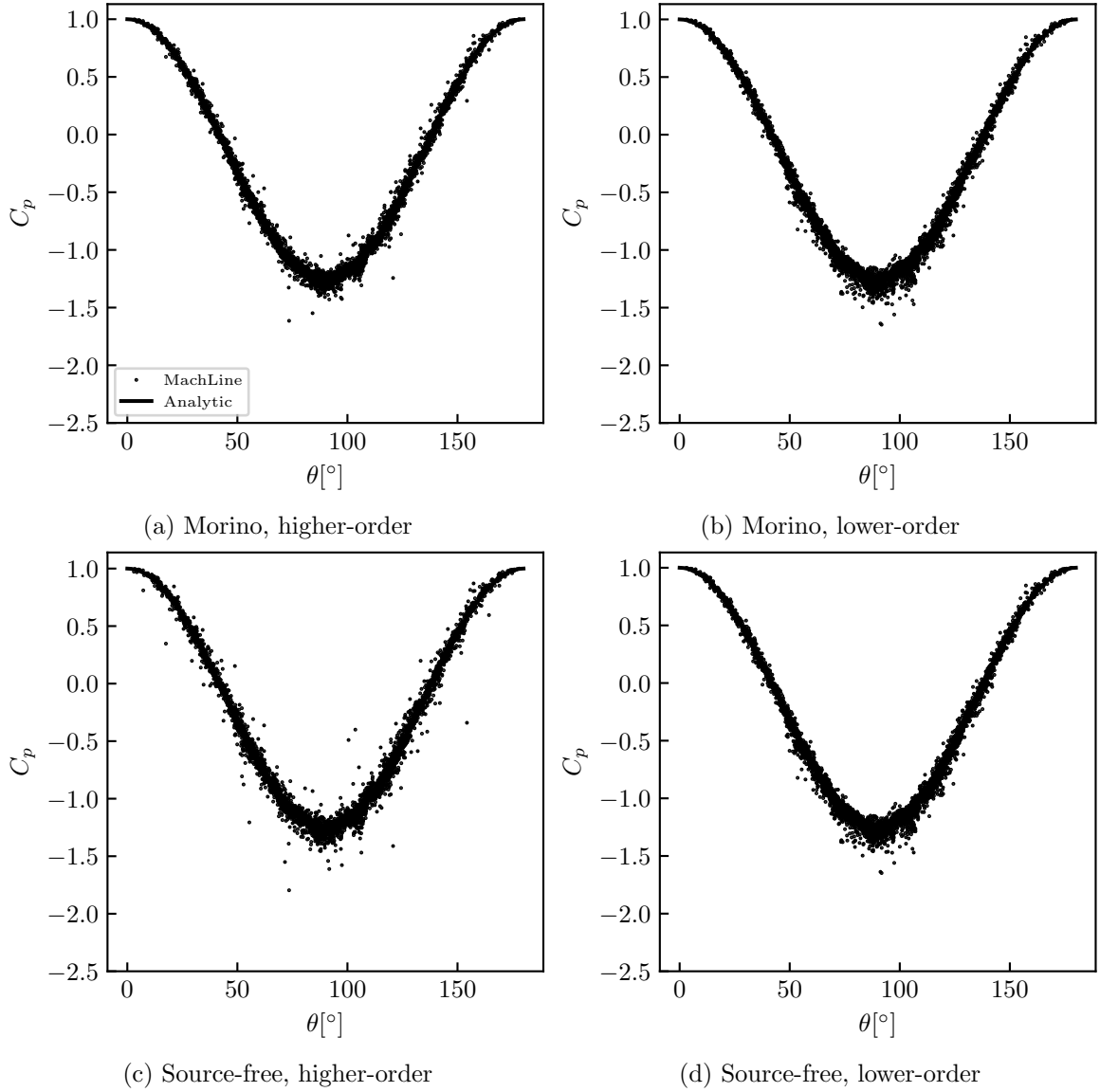


Fig. 8.18: Pressure coefficient predictions for a fine, randomly-paneled sphere in incompressible flow compared to the analytic result.

the spindle (the two tip vertices were placed exactly) and then generated panels between the vertices using a convex hull algorithm. An example mesh with 800 vertices is shown in Fig. 8.19.

The isentropic pressure results from MachLine were compared against a solution obtained using the method of characteristics at $M_\infty = \sqrt{2}$, as reported by [2]. The method of characteristics uses a nonlinear flow model, rather than the linearized Prandtl-Glauert

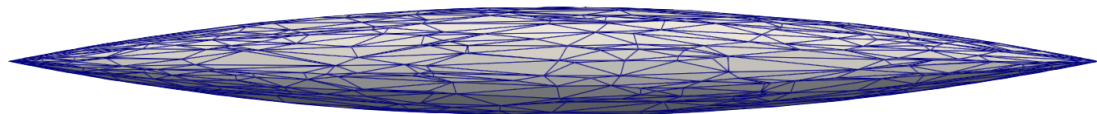


Fig. 8.19: Example mesh for the randomly-paneled spindle with 800 vertices.

equation. Because of this, differences between MachLine and the analytic results were due to both differences in the flow model and the discretization used in MachLine. However, since the Mach number was low and the spindle was fine, it was assumed that errors due to differences in the model were small relative to the discretization errors.

Fig. 8.20 shows the results of this comparison. Over the center portion of the spindle, the results from MachLine matched the analytic solution quite well for all formulations and singularity distributions. There were more erroneous pressure results when higher-order distributions were used than when lower-order distributions were used, showing the higher-order distributions had greater mesh sensitivity in this case. This was different than the comparison for a randomly-paneled sphere in incompressible flow. In that case, the Morino formulation paired with the higher-order distributions showed the least sensitivity to panel regularity. This is likely due to the difference in flow character. In supersonic flow, disturbances do not propagate upstream, but within MachLine there is some upstream influence created due to how the higher-order source and doublet fits are formulated (see Section 5.5). In PAN AIR, these fits were weighted to minimize upstream influence [2, 33]. It may be possible to weight the fits described here to mimic this effect. This is an area for potential future research.

In all cases, the pressure predictions were particularly poor at the trailing tip. It should be noted that Ehler's et al. reported similar behavior with PAN AIR [2]. All of the panels

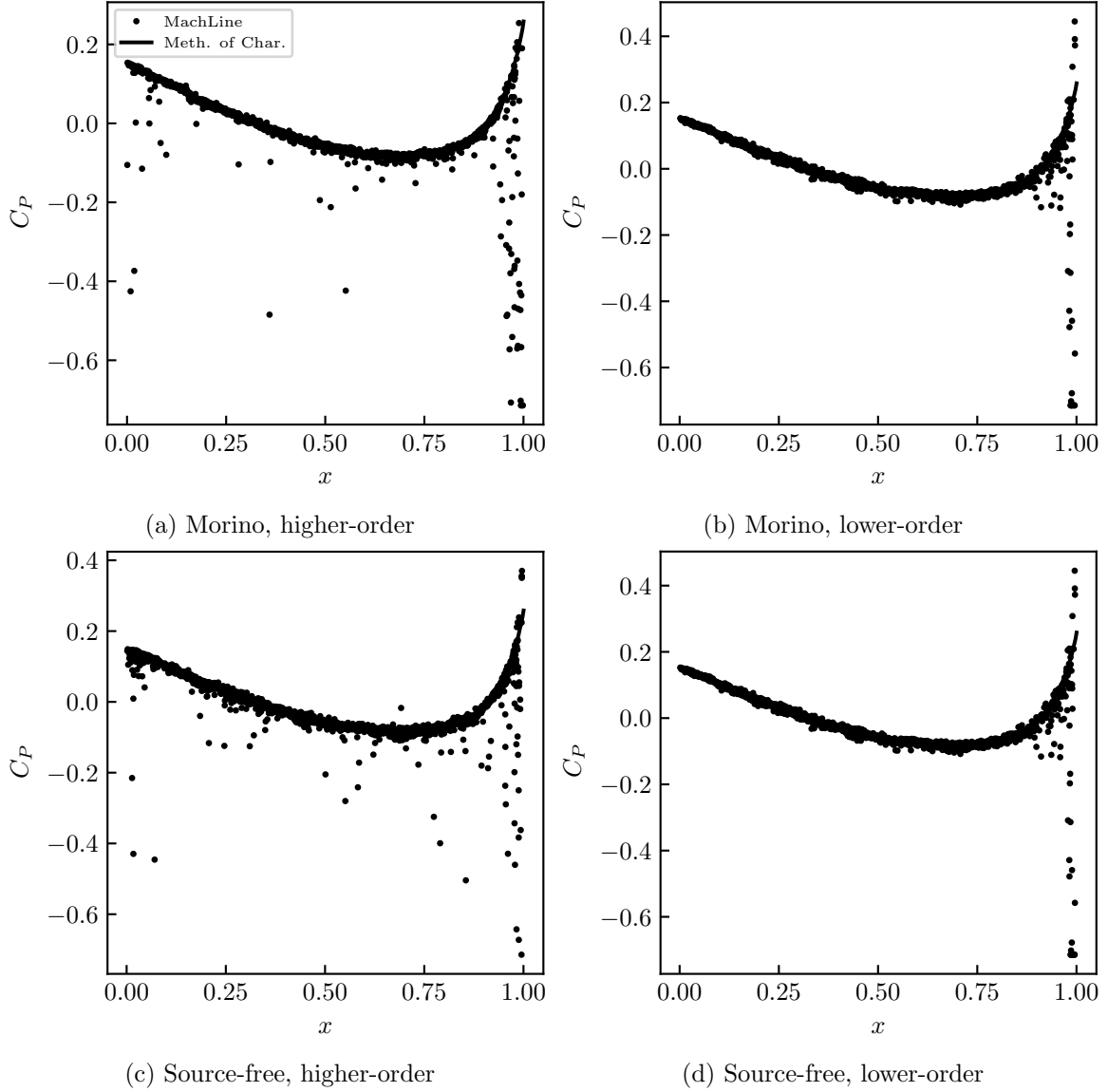


Fig. 8.20: Pressure coefficient predictions for a randomly-paneled spindle in supersonic flow compared to results from the method of characteristics [2].

on the spindle varied greatly in aspect ratio, length, and inclination. However, the results at the tip appear to be particularly sensitive to these variations. For this reason, it is highly recommended to enforce regular paneling at the tips of axisymmetric bodies (c.f. the results in Section 9.4).

8.3.3 Discussion

Overall, the lower-order distributions resulted in the least sensitivity to panel regularity for both subsonic and supersonic flows. For subsonic flows, the higher-order distributions paired with the Morino formulation resulted in the least sensitivity to panel regularity; however, this was not the case for supersonic flows, likely due to how some upstream influences were introduced by using the higher-order distributions in supersonic flow. As evidenced by the spindle in supersonic flow, it is particularly important for the paneling at pointed tips to be regular.

CHAPTER 9

MODELING STUDIES

Essentially, all models are wrong, but some are useful. However, the approximate nature of the model must always be borne in mind... - George Box

Having examined the numerical behavior of the pilot code, its accuracy may now be evaluated through various case studies. In most of these studies, the accuracy of the pilot code compared to analytic flow solutions or wind tunnel data is evaluated. These studies should serve as a reference for future users and developers of MachLine.

As with the numerical behavior of the method, the effect of boundary condition formulation and singularity distribution on the accuracy of solutions is here examined. As before, the different combinations of these are labeled as listed in Table 8.1.

9.1 Sphere in Compressible Subsonic Flow

Tamada [3, 142] presented a second-order accurate analytic solution for compressible, sub-critical flow over a sphere. Predictions from MachLine are here compared to Tamada's solutions for $M_\infty = 0.4$ and 0.5 . This was done using the direct subsonic calculations. Comparison is made here only between predicted velocity magnitudes, and so the method of correcting for subsonic flows using the pressure corrections listed in Section 7.2.3 was not considered.

The results for $M_\infty = 0.4$ and 0.5 are shown in Figs. 9.1 and 9.2, respectively. In each case, the freestream velocity was unity. For $M_\infty = 0.4$, all boundary condition formulations and singularity distributions resulted in surface velocity distributions that matched Tamada's solution very closely. The highest discrepancies appeared toward the front of the sphere. This would be expected, as this was the stagnation region. At stagnation, the flow has been significantly perturbed from the freestream, and so the linear Prandtl-Glauert equation was less valid there.

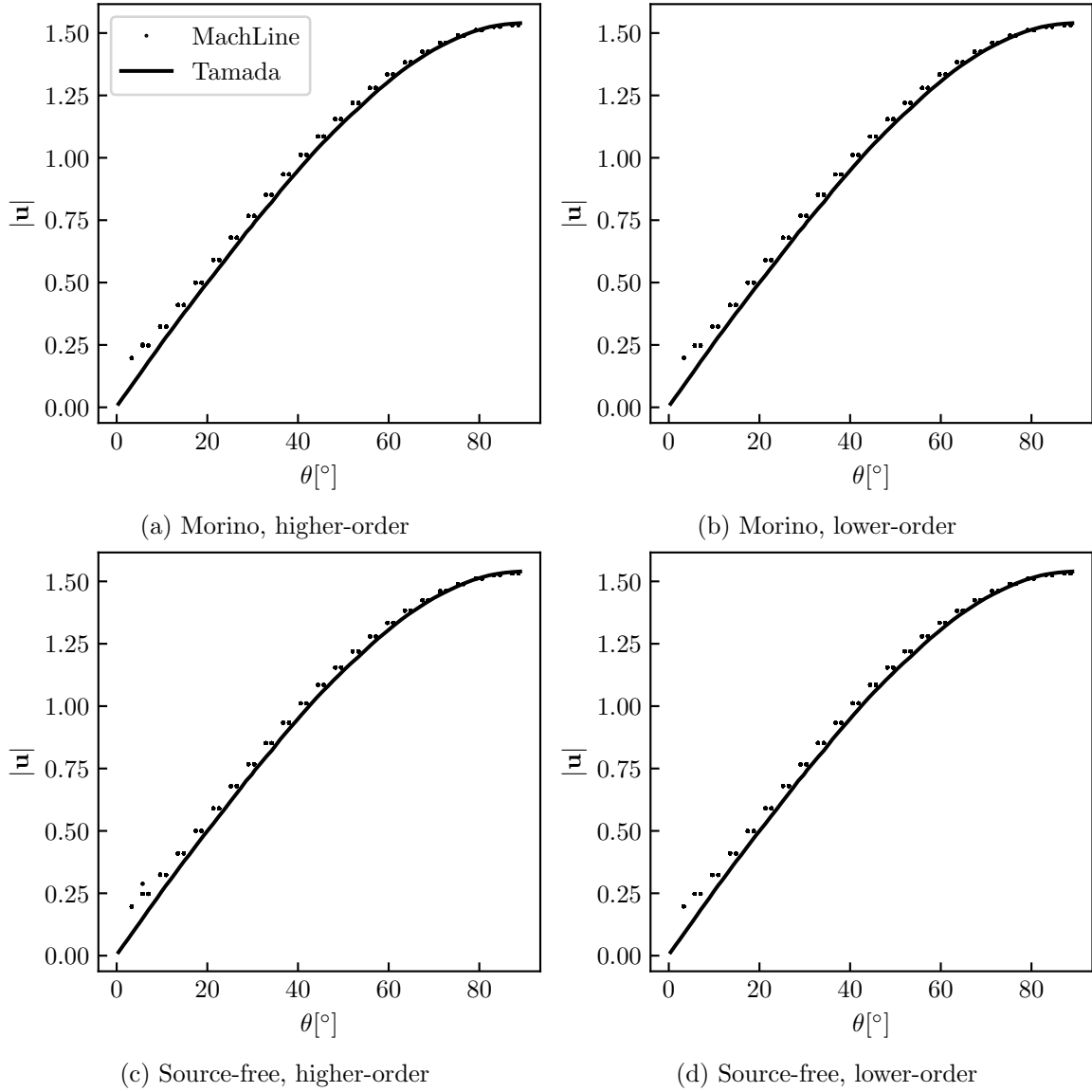


Fig. 9.1: Surface velocity predictions over the front half of a sphere at $M_\infty = 0.4$ compared to the second-order result of Tamada [3].

At $M_\infty = 0.5$, significant discrepancies appeared between Tamada's solution and the MachLine results over most of the sphere. At this higher Mach number, the flow was closer to transonic, and the flow perturbations were greater, making the linear Prandtl-Glauert equation even less valid than at $M_\infty = 0.4$. These cases highlight the need for caution when using the compressible Prandtl-Glauert equation to model flows with large perturbations from the freestream.

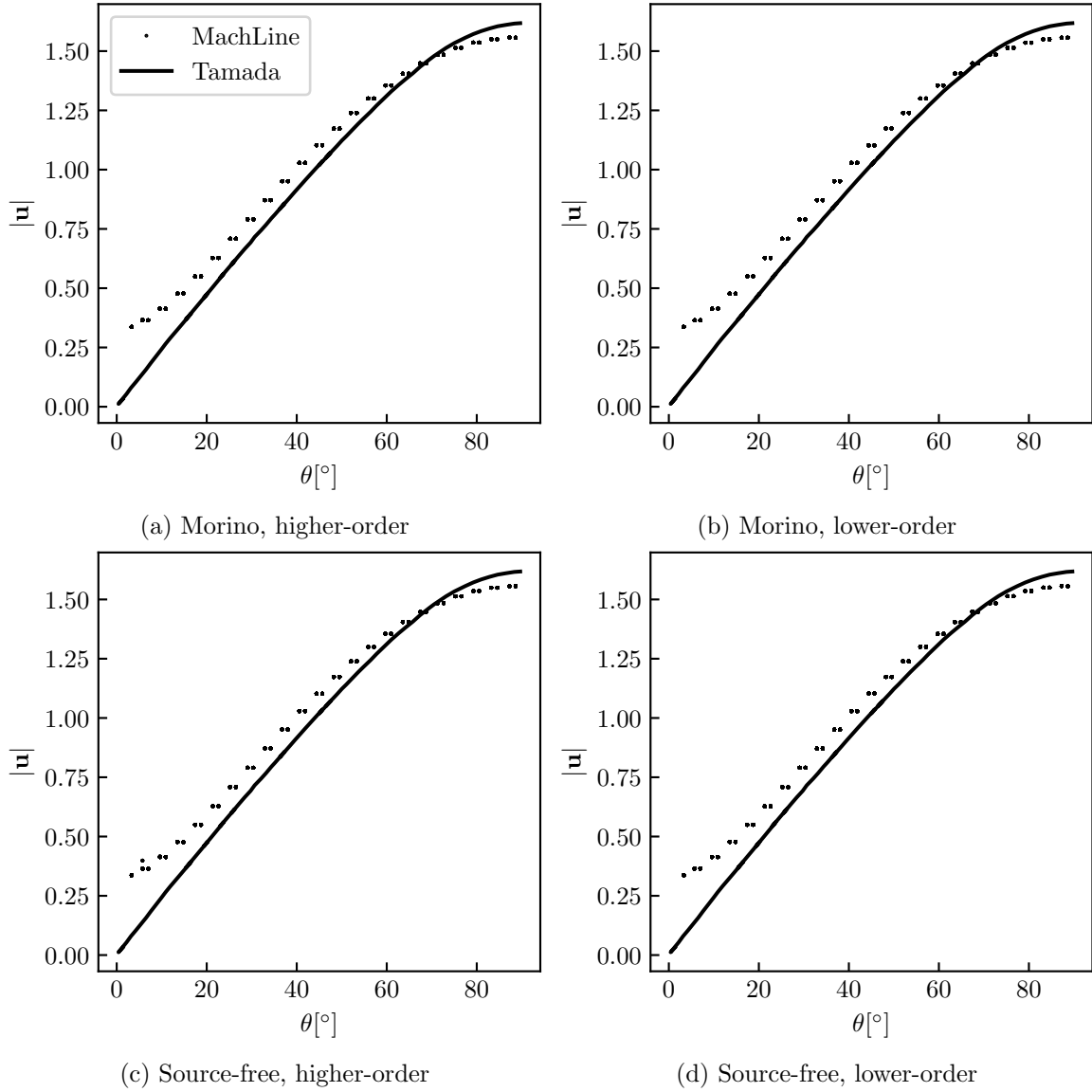


Fig. 9.2: Surface velocity predictions over the front half of a sphere at $M_\infty = 0.5$ compared to the second-order result of Tamada [3].

Another source of these discrepancies was the use of a linearized mass flux boundary condition. At the very front of the sphere at $M_\infty = 0.5$, the normal component of velocity predicted by MachLine was approximately a third of the freestream velocity, rather than 0 as would be expected for an impermeable surface. However, due to the linearized mass flux being used, the velocities were never forced to go to zero here. This highlights the approximate nature of the linearized mass flux boundary condition for non-slender geometries.

For this simple case with a very regular mesh, there were no visible differences in the results between singularity distributions and boundary condition formulations. However, there were noticeable differences in the time required to compute each solution. These times are listed in Table 9.1. For almost all cases, using the source-free boundary condition formulation resulted in faster computation than using the Morino formulation, as would be expected. In addition, using the higher-order singularity distributions typically increased the computation time compared to the lower-order distributions, as would also be expected. As the size of the computational mesh increased, the differences in computation time due to order of singularity distribution became much greater than those due to boundary condition formulation.

Table 9.1: Total computation run times in seconds averaged across freestream conditions for the sphere in compressible flow.

	ML	MH	SL	SH
Ultra Coarse	0.025	0.024	0.017	0.018
Coarse	0.067	0.049	0.045	0.045
Medium	0.310	0.358	0.294	0.338
Fine	3.820	4.864	3.741	4.489

9.2 ONERA M6 Wing at High-Subsonic Mach Numbers

The ONERA M6 wing was a research model developed for evaluating numerical models of compressible subsonic and transonic flows [4]. The M6 wing had moderate aspect ratio, a symmetric airfoil section, and was slightly tapered and swept. The experimental results compared with in this study were reported by Schmitt and Charpin [4]. Schmitt and Charpin tested a model of the ONERA wing at high subsonic Mach numbers and reported pressure distributions at various spanwise locations.

The airfoil section as reported by Schmitt and Charpin required some minor modifications in order to produce good computational results. The airfoil outline points as reported in [4] resulted in oscillations in the flow results. In addition, the trailing edge of the wing was blunt, which can cause issues in a panel method. In a later study by Mayeur et al. [143]

using high-fidelity CFD, these issues were resolved by smoothing the airfoil outline and linearly extending the trailing edge until it was sharp. The smoothed and extended airfoil points reported by Mayeur et al. were used here.

Since the experimental data were obtained at compressible subsonic Mach numbers ($M_0 \approx 0.7$ and $M_0 \approx 0.84$), it was possible to predict the pressure distributions using both the direct subsonic calculation and the Prandtl-Glauert correction (see Section 7.2.3) within MachLine. The Prandtl-Glauert correction is often used in modeling subsonic flows over three-dimensional configurations, despite the fact that it is derived from two-dimensional flow theory. Having both the Prandtl-Glauert correction and direct subsonic calculations implemented in MachLine makes it possible to test the validity of this application of the Prandtl-Glauert correction.

Within MachLine, three different levels of mesh refinement were used. Between the medium and fine levels, the differences in surface pressures were minimal. The results were thus deemed sufficiently converged, and all results reported here were obtained using the finest mesh.

Figure 9.3 shows the results at 20% semispan, $M_\infty = 0.7$, and $\alpha = 0^\circ$. Both the direct subsonic calculation and the Prandtl-Glauert correction gave results that followed the experimental data well, with the direct subsonic calculation being slightly more accurate in general. At this small angle of attack, the perturbations to the flow induced by the wing were small. Being near the root, the flow over this section was mostly two-dimensional. Because of these facts, it makes sense that the predictions from the direct calculations and the Prandtl-Glauert correction were very similar. Except for the source-free formulation paired with the higher-order singularity distributions, the results obtained from the different boundary condition formulations and singularity distributions appeared identical.

The source-free formulation paired with higher-order singularity distributions resulted in unrealistic spikes in pressure at both the leading and trailing edges. These spikes were much larger when the Prandtl-Glauert correction was used. The fact that the pressure spikes coming from the direct calculation were smaller was likely due to the direct calculation

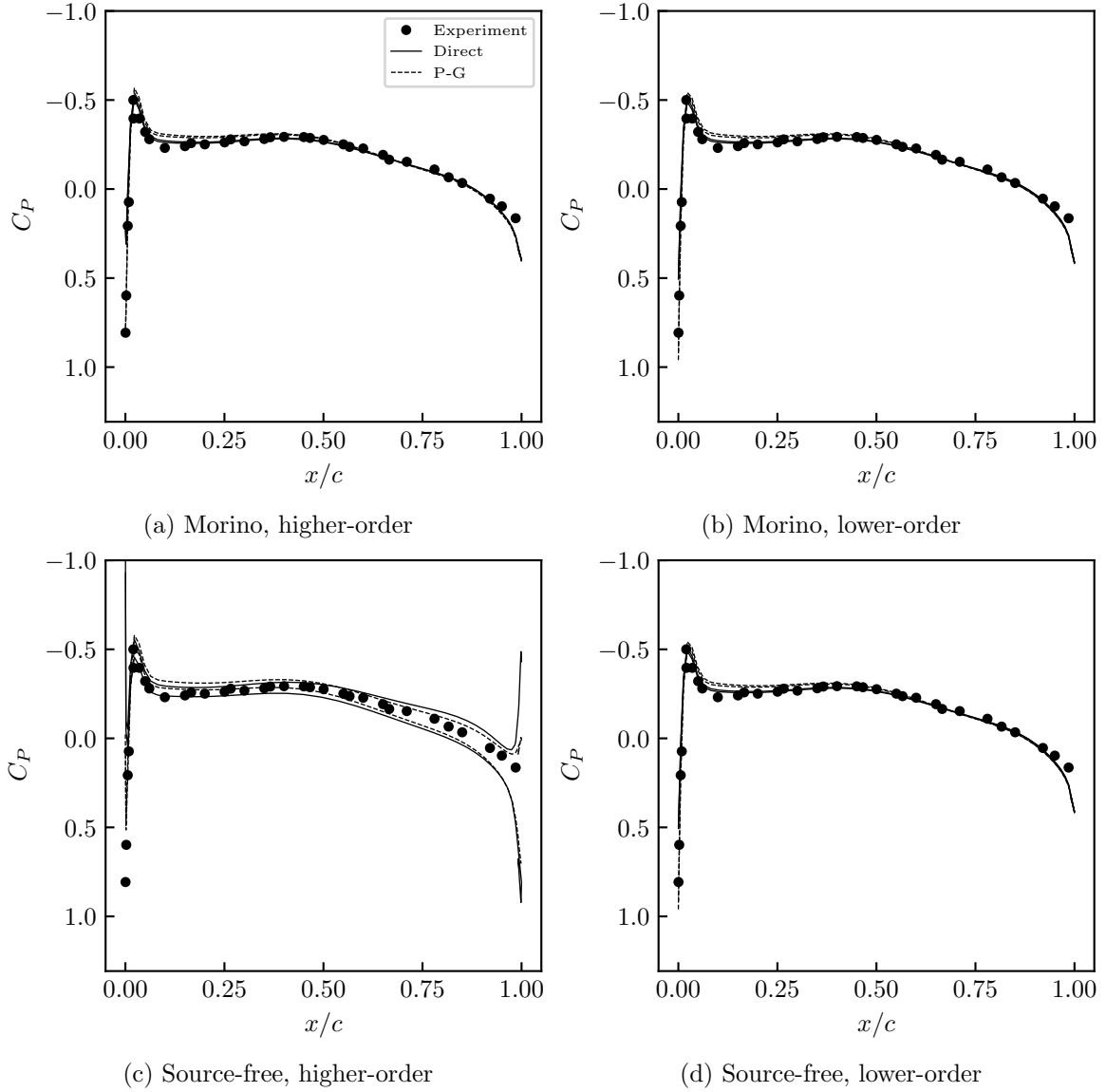


Fig. 9.3: Results at 20% semispan on the ONERA M6 wing at $M_\infty = 0.7$ and $\alpha = 0.06^\circ$ from MachLine using both the direct subsonic calculation and the Prandtl-Glauert correction. Experimental results are included from [4].

method using the isentropic pressure coefficient, which has hard upper and lower limits. There is no lower limit to the incompressible pressure coefficient, on which the Prandtl-Glauert pressure correction is based. Notice the lower spike seen on the trailing edge was the same for both the direct calculation and the Prandtl-Glauert correction. Outside of these spikes, the pressure distributions predicted using the source-free formulation and

higher-order distributions were still reasonably accurate, though there was a noticeable difference between the upper and lower surface pressures. These spikes were not due to an ill-conditioned system; the condition number of the AIC matrix was never larger than 1.5×10^5 for the most-refined mesh.

The pressure spikes at the trailing edge may be seen more clearly in Fig. 9.4. Interestingly, these spikes only appeared for the finest mesh and only with the source-free boundary condition formulation and higher-order distributions. Since these spikes were clustered at the trailing edge, they may be due to how the discontinuous panel edges at the trailing edge were handled. Using the midpoint-interpolation method, the doublet strength along these edges was forced to be linear, whereas all other edges in the mesh could have quadratic distributions of doublet strength. On the finest mesh, this shift had to occur over a much shorter distance than on the other meshes. This sudden shift from a quadratic to linear distribution may have been the cause of the poor pressure results.

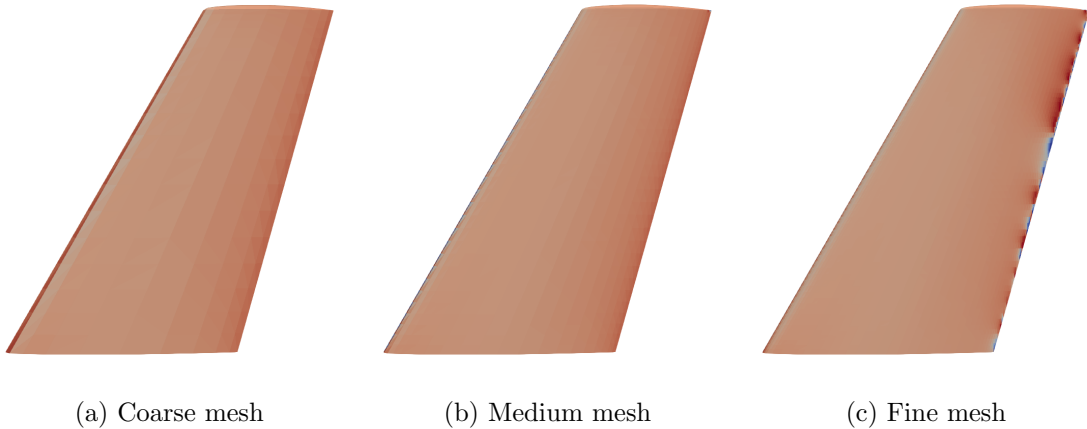


Fig. 9.4: Top-down view of the isentropic pressure distribution over the ONERA M6 wing at $M_\infty = 0.7$ and $\alpha = 0.06^\circ$ using varying mesh resolutions. Color scale ranges from $C_P \approx 1.1$ (dark red) to $C_P \approx -2.9$ (dark blue).

The leading-edge pressure spikes were different in nature than the trailing-edge spikes. A head-on view of the leading edge at 0° angle of attack is shown in Fig. 9.5. At 0° angle of attack, the local fluid velocity should be lowest at the leading edge, resulting in a high

pressure coefficient. This is not what is shown in Fig. 9.5. For the results predicted here, the set of panels abutting the leading edge had a very low pressure coefficient compared to the surrounding panels with a jump in pressure coefficient between panels on the order of 1.5. This may have been due to the shape and arrangement of panels at the leading edge. At the leading edge, the panels flip in orientation, leading to three neighboring vertices used to interpolate the midpoint doublet strengths lying nearly on a line. This was the cause of the six-point-fit method's failure and may be a source of mesh sensitivity for the midpoint-interpolation method.

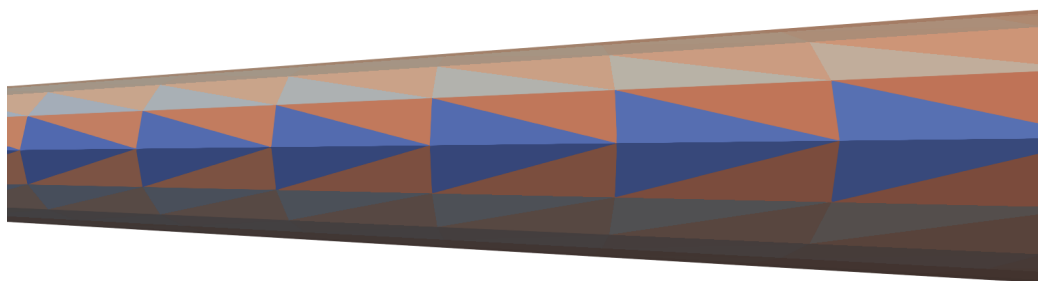


Fig. 9.5: View of the leading edges of the ONERA M6 wing at 0° angle of attack, facing downstream, at the medium mesh resolution with higher-order singularity distributions. Color scale ranges from approximately $C_P \approx -1.6$ (blue) to $C_P \approx 0.1$ (red).

The differences between the direct subsonic calculations and the Prandtl-Glauert correction increased with angle of attack. Figure 9.6 shows the results at 90% semispan, $M_\infty = 0.7$, and $\alpha = 5^\circ$. Here, the Prandtl-Glauert correction overestimated the magnitude of the pressure coefficient on the upper surface of the wing. The two methods produced very similar results on the lower surface of the wing. At this flow condition, the different boundary condition formulations and singularity distributions had even less of an effect on the results. Only in the case of the source-free boundary condition formulation paired with the higher-order singularity distributions were there still spikes in pressure at the trailing edge. Again, these spikes were much worse when the Prandtl-Glauert correction was used.

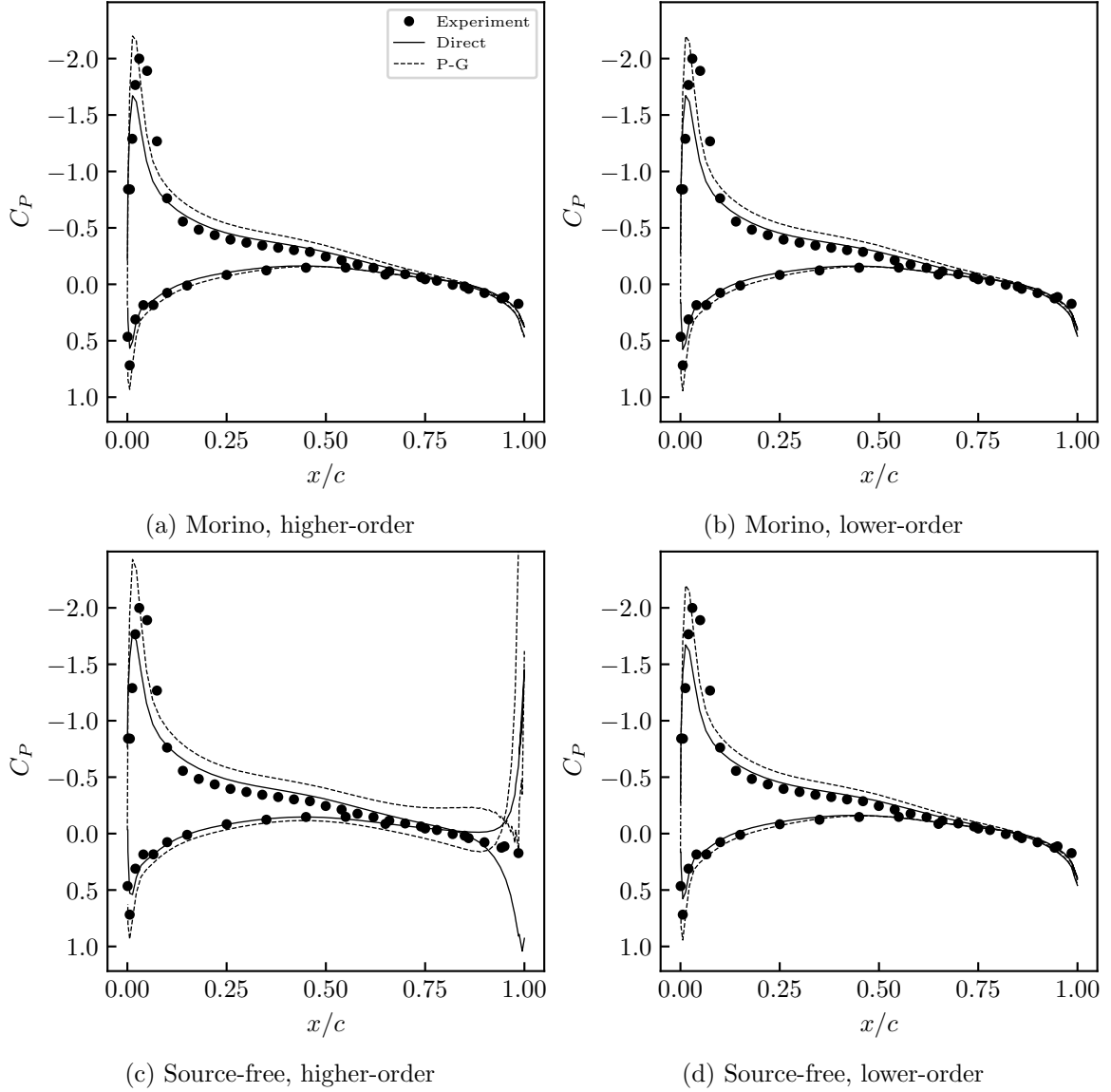


Fig. 9.6: Results at 90% semispan on the ONERA M6 wing at $M_\infty = 0.7$ and $\alpha = 5^\circ$ from MachLine using both the direct subsonic calculation and the Prandtl-Glauert correction. Experimental results are included from [4].

Near the tip of the wing, the flow was more three-dimensional, and so larger differences between the direct calculations and the Prandtl-Glauert correction would be expected there. The results at 99% semispan, $M_\infty = 0.7$, and $\alpha = 5$ are shown in Fig. 9.7. Here, there were significant deviations in the predicted results from the experimental results, particularly over the trailing half of the wing. On the trailing half of the wing, both the direct

calculations and the Prandtl-Glauert correction resulted in predicted pressures that were significantly higher than the experimental results. In addition, the results from MachLine did not show the pressure coefficient matching at the trailing edge, as would be expected from the Kutta condition. This was likely due to the simplified way in which the Kutta condition is enforced in MachLine. Improving this implementation is an area of potential future research. Between all combinations of boundary condition formulations and singularity distributions, there were minimal differences in the predicted pressures, aside from spikes at the trailing edge resulting from the source-free formulation and higher-order distributions. Surprisingly, the differences between the Prandtl-Glauert correction and the direct subsonic calculations were not large. Compared with the experimental data, neither predicted the three-dimensional flow at this section of the wing very well.

The pressure results at 90% semispan, $M_\infty = 0.84$, and $\alpha = 0$ are shown in Fig. 9.8. At this higher Mach number, the flow was clearly supercritical. There was a locally supersonic bubble beginning at about 10% chord and a small shockwave at about 25% chord. MachLine did not capture this shock at all using either calculation method.

In the locally-supersonic region seen in Fig. 9.8, the governing differential equation changed from elliptic to hyperbolic. Since the Prandtl-Glauert equation assumes a constant Mach number, this local behavior cannot be modeled within a linear panel method. Thus, the results from MachLine for the supercritical case were as expected. Even so, Crovato et al. [116] suggest that linear panel methods are still useful at supercritical Mach numbers for predicting lift-induced loads for wing design. It is really in the area of supercritical/transonic wave drag prediction where linear panel methods fall short.

The total method run times for the ONERA wing are given in Table 9.2. As was seen with the sphere in compressible flow, the higher-order singularity distributions consistently resulted in greater run times than the lower-order distributions. The differences between the boundary condition formulations here were minimal, with the source-free formulation generally running faster for the coarser meshes.

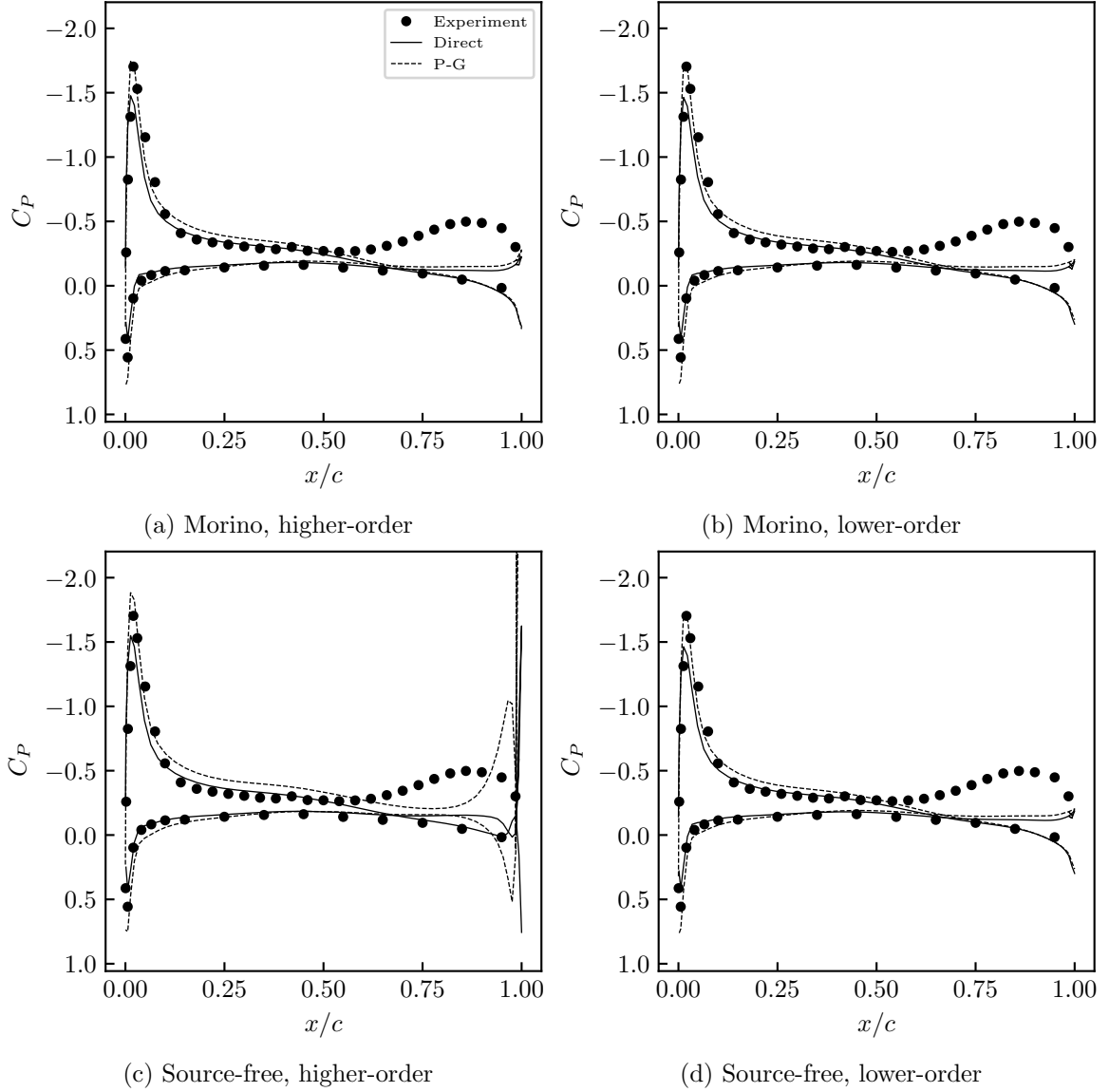


Fig. 9.7: Results at 99% semispan on the ONERA M6 wing at $M_\infty = 0.7$ and $\alpha = 5^\circ$ from MachLine using both the direct subsonic calculation and the Prandtl-Glauert correction. Experimental results are included from [4].

Table 9.2: Total computation run times in seconds averaged across freestream conditions for the ONERA M6 wing.

	ML	MH	SL	SH
Coarse	0.322	0.355	0.325	0.345
Medium	5.191	5.776	5.155	5.575
Fine	87.089	97.589	88.436	98.528

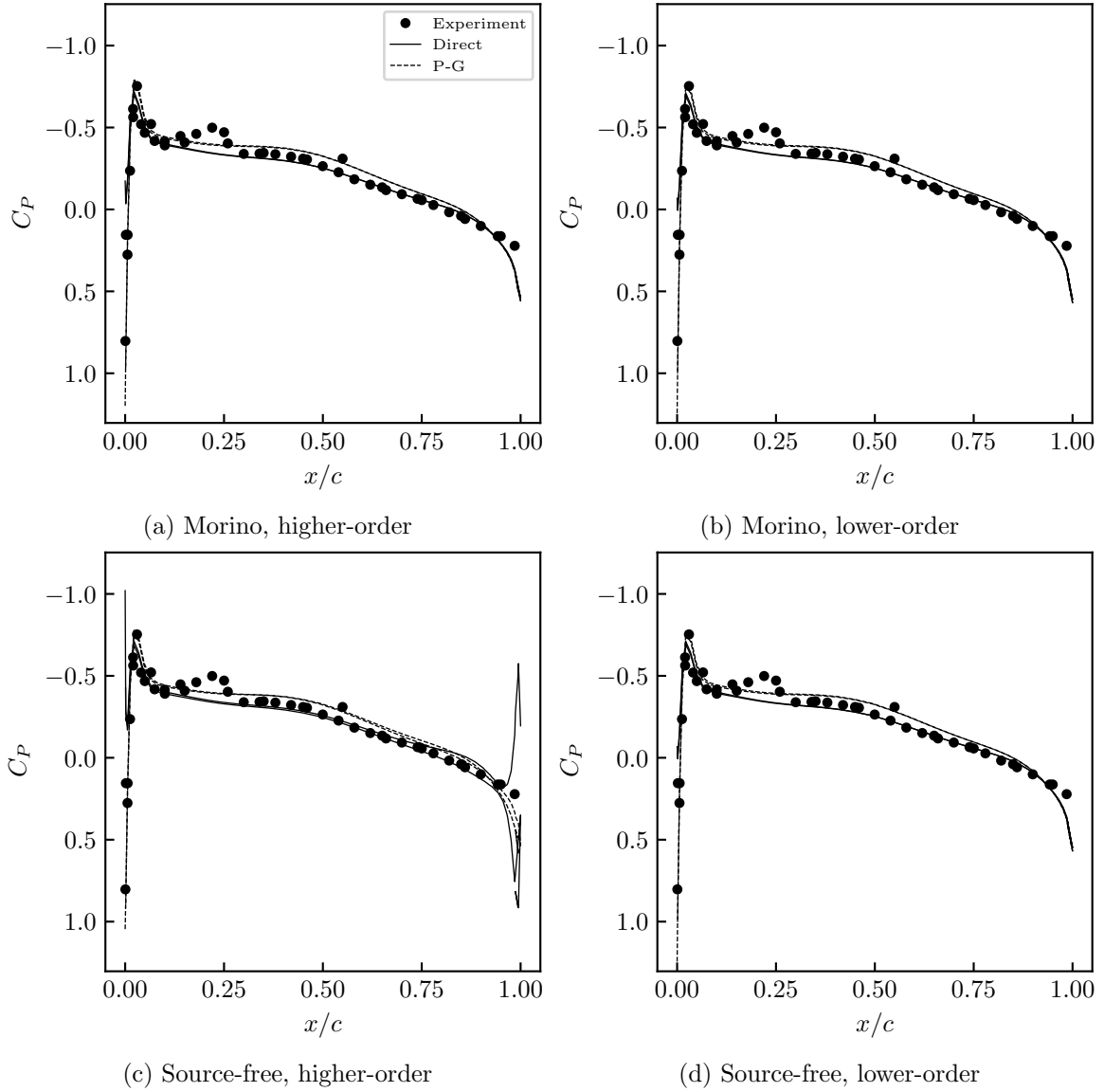


Fig. 9.8: Results at 90% semispan on the ONERA M6 wing at $M_\infty = 0.84$ and $\alpha = 0^\circ$ from MachLine using both the direct subsonic calculation and the Prandtl-Glauert correction. Experimental results are included from [4].

9.3 Diamond (Double-Wedge) Airfoil

Having examined two standard subsonic flow cases, the remainder of this work focuses on supersonic flows. The first supersonic case considered here was a straight wing with a diamond airfoil section. For straight, untapered wings with large aspect ratio in supersonic flow, the flow in the root region of the wing is two-dimensional and may be analyzed using

shock-expansion theory [141]. This is done primarily on wings with wedge shapes, as shock-expansion theory assumes the body being analyzed may be broken up into a finite number of flat surfaces. Over each of these surfaces, the pressure coefficient is determined as a function of the surface inclination and upstream flow conditions. From this analysis, the pressure coefficient distribution over any flat surface is constant. Shock-expansion theory is exact for inviscid flows, with no limitations on the magnitude of perturbations so long as the shocks remain attached.

The wing considered here had a wedge half-angle of 5° (see Fig. 9.9) and a span large enough such that the entire root of the wing remained outside the tip Mach cones for all Mach numbers considered. A maximum continuity angle of 1° was used (see Section 8.2.3 for a discussion on the impact of this). The same meshes were used for this study as were used for the study in Section 8.2.3. Based on a mesh sensitivity analysis, it was determined that the surface pressures considered in this study (i.e. those along the root) were insensitive to the mesh density. The greatest variation in pressure coefficient at the root of the wing was on the order of 10^{-3} between the most- and least-refined meshes. As such, the most-refined mesh was determined to be sufficiently converged, and the results presented here were obtained using that mesh.

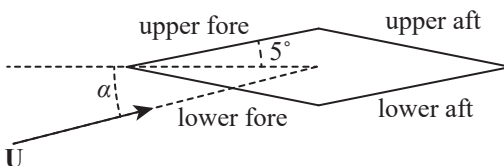


Fig. 9.9: Airfoil section used for the diamond wing with surfaces labeled.

The results for $M_\infty = 1.5$ are shown in Fig. 9.10. Each line and the discrete points clustered around it represent the pressure coefficient on each surface of the double-wedge section as labeled in Fig. 9.9. The different marker shapes are the different pressure rules

used in MachLine. At lower angles of attack, the pressure rules were all very close, but the differences between the pressure rules increased with angle of attack. This is expected, as at higher angles of attack, the small-perturbation assumptions built into some of the pressure rules become less valid. The predictions from MachLine were also very close to those from shock-expansion theory. The largest differences between the results from MachLine and those from shock-expansion theory appeared on the lower-fore and upper-aft surfaces at higher angles of attack. These surfaces were more inclined to the freestream than the other surfaces and so had greater flow perturbations. Conversely, for the upper-fore and lower-aft surfaces, the pressure predictions from MachLine improved with increasing angle of attack. At 5° angle of attack, these surfaces were aligned with the freestream, and MachLine predicted pressure coefficients very close to zero, as would be expected.

At the highest angles of attack, where the results from MachLine were the furthest from the shock-expansion results, there was also the greatest variation between the results obtained using different pressure rules. This validates the assertion made in Sec. 7.4 that the relative accuracy of the panel method varies inversely with the spread of the different pressure rules.

At all angles of attack and for all surfaces, there were no noticeable differences between the different boundary condition formulations and singularity distributions. This makes sense as the pressure over each surface should be constant. Constant pressure distributions come from linear distributions of doublets and constant distributions of sources. Because of this, the higher-order singularity distributions likely ended up with the higher-order singularity distribution parameters (σ_ξ , $\mu_{\xi\xi}$, etc.) simply being zero. The only thing that would likely introduce higher-order effects would be the break at the line of maximum thickness. However, because the maximum continuity angle was set to 1° , the line of maximum thickness was marked as discontinuous, and the higher-order distributions were there forced to be linear.

The results for $M_\infty = 2$ are shown in Fig. 9.11. These results were very similar to those at $M_\infty = 1.5$ except that the differences between pressure rules were greater for the

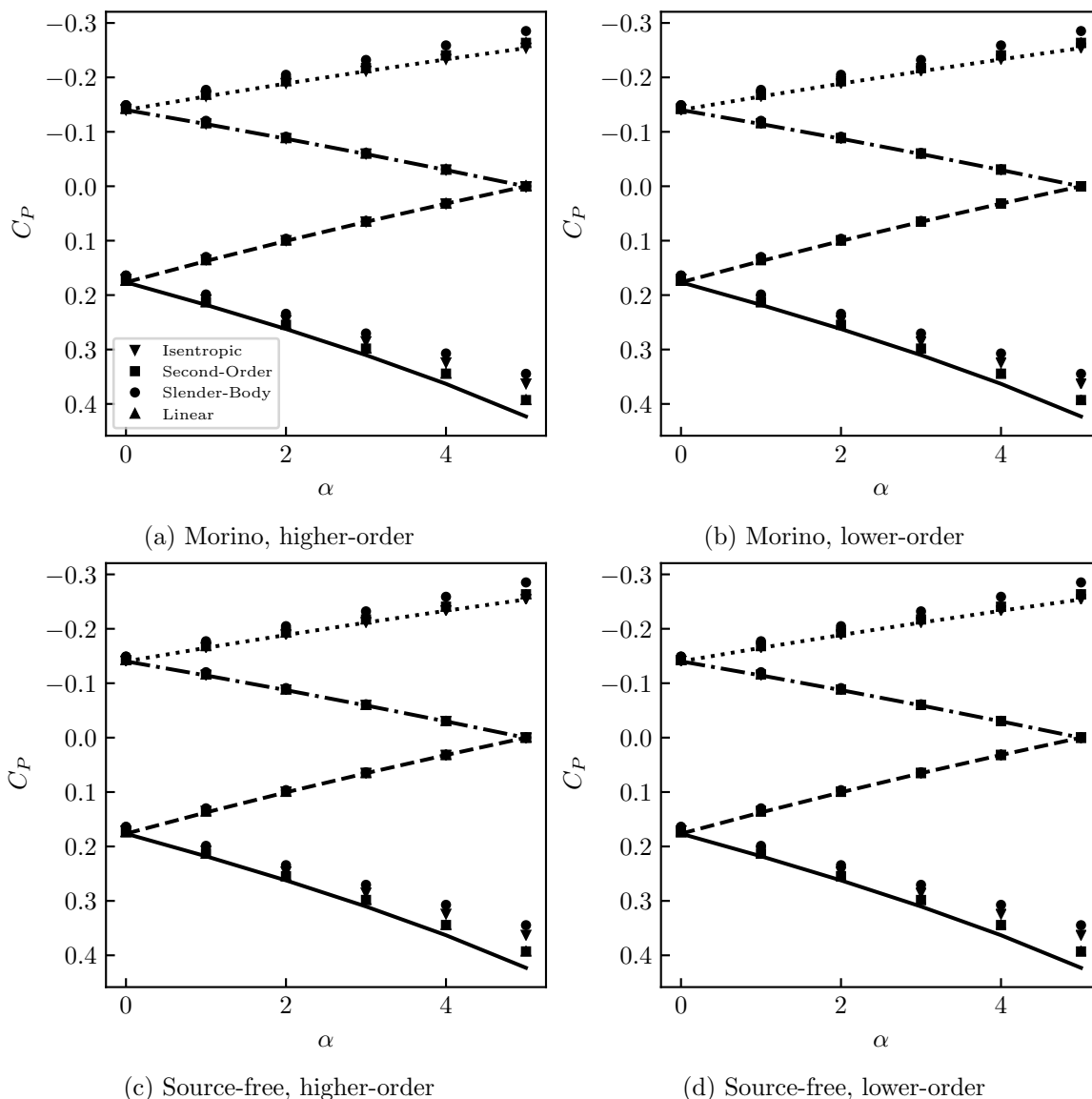


Fig. 9.10: Surface pressures for a 5° half-angle double-wedge airfoil over angle of attack at $M_\infty = 1.5$ predicted by MachLine (discrete points) and shock-expansion theory (lines). The line style indicates which surface the pressure coefficient is over (see Fig. 9.9). Solid: lower-fore. Dashed: upper-fore. Dash-dot: lower-aft. Dotted: upper-aft.

lower-forward and upper-aft surfaces at higher angles of attack. As discussed before, the linearizing assumptions made to obtain the Prandtl-Glauert equation become less valid with increasing Mach number. Despite this, the pressure predictions from MachLine were still very good, with the isentropic pressure rule matching shock-expansion theory even out to 5° angle of attack.

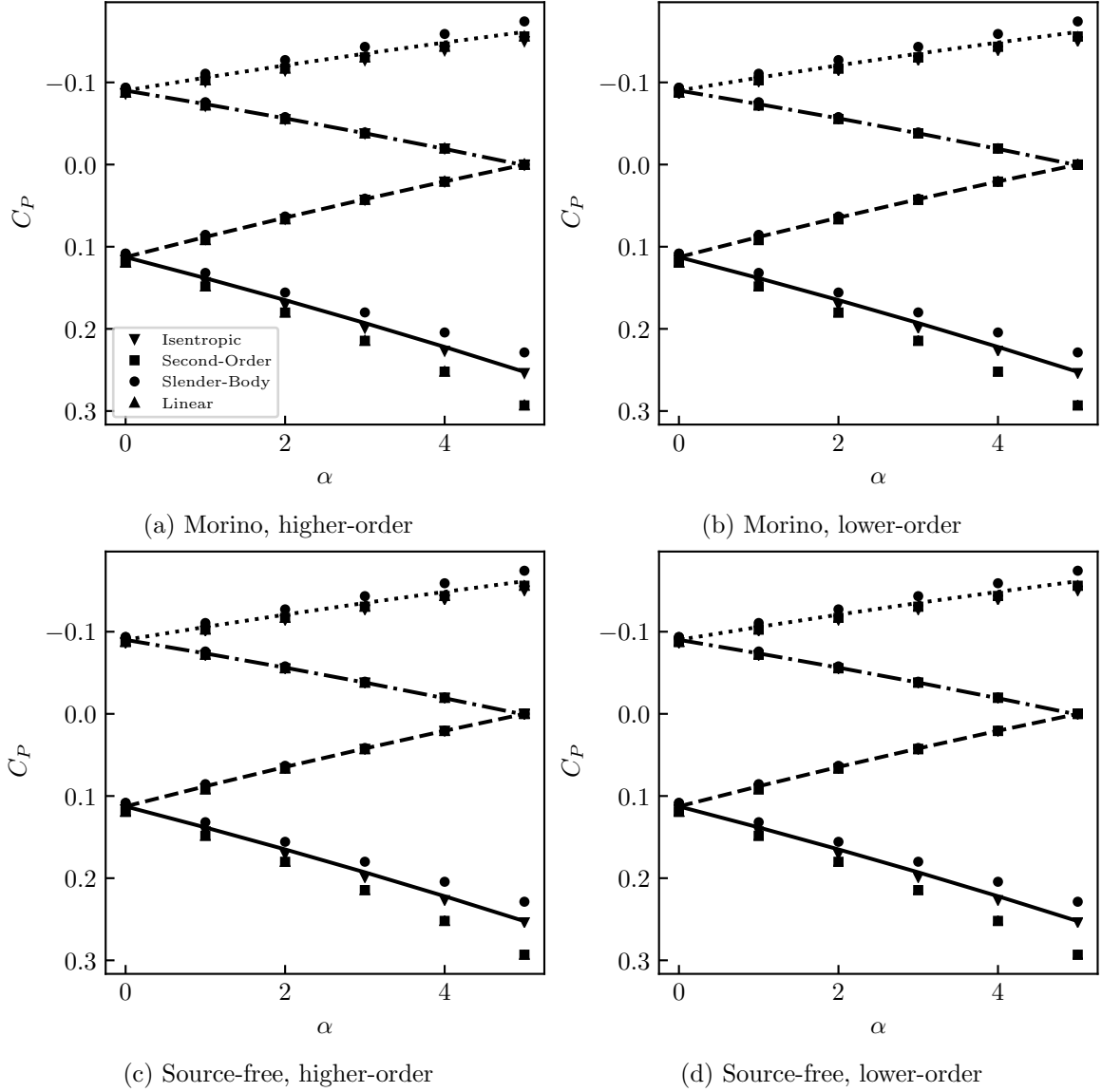


Fig. 9.11: Surface pressures for a 5° half-angle double-wedge airfoil over angle of attack at $M_\infty = 2$ predicted by MachLine (discrete points) and shock-expansion theory (lines). The line style indicates which surface the pressure coefficient is over (see Fig. 9.9). Solid: lower-fore. Dashed: upper-fore. Dash-dot: lower-aft. Dotted: upper-aft.

The results from $M_\infty = 3$ are shown in Fig. 9.12. These results were qualitatively similar to the results at $M_\infty = 1.5$ and $M_\infty = 2$. At this higher Mach number, the predictions from MachLine began to diverge from shock-expansion theory more quickly than before. Again, the isentropic pressure rule tended to produce the most accurate predictions,

and there were still no noticeable differences between the boundary condition formulations and singularity distributions.

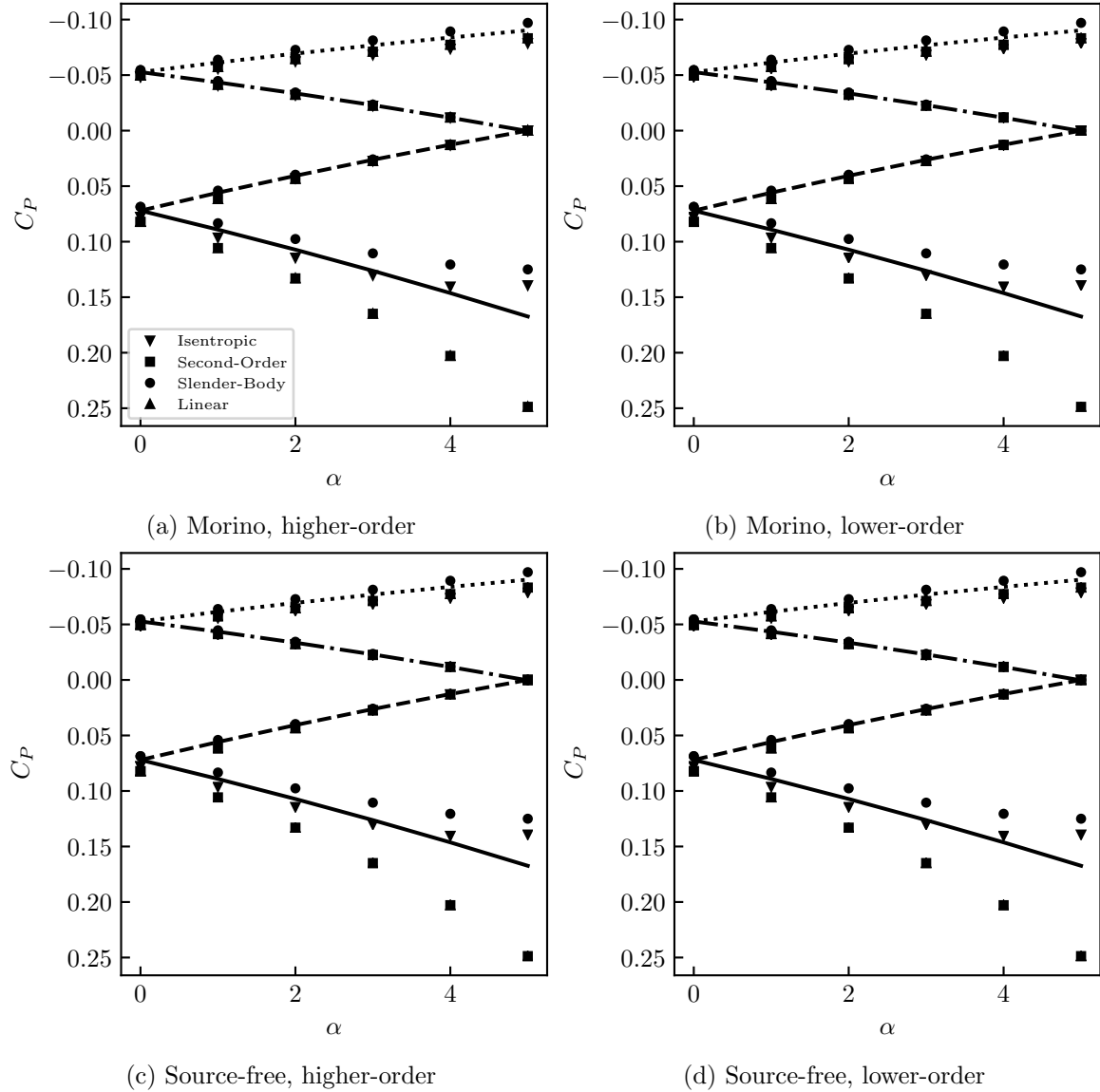


Fig. 9.12: Surface pressures for a 5° half-angle double-wedge airfoil over angle of attack at $M_\infty = 3$ predicted by MachLine (discrete points) and shock-expansion theory (lines). The line style indicates which surface the pressure coefficient is over (see Fig. 9.9). Solid: lower-fore. Dashed: upper-fore. Dash-dot: lower-aft. Dotted: upper-aft.

The results for $M_\infty = 5$ are shown in Fig. 9.13. Here, the results from MachLine diverged from shock-expansion theory almost immediately, though the predictions from

MachLine were still very accurate at 0° angle of attack. In addition, it was only on the lower-forward surface that the predictions from MachLine significantly diverged. On all other surfaces, the pressure predictions from MachLine remained consistent and accurate.

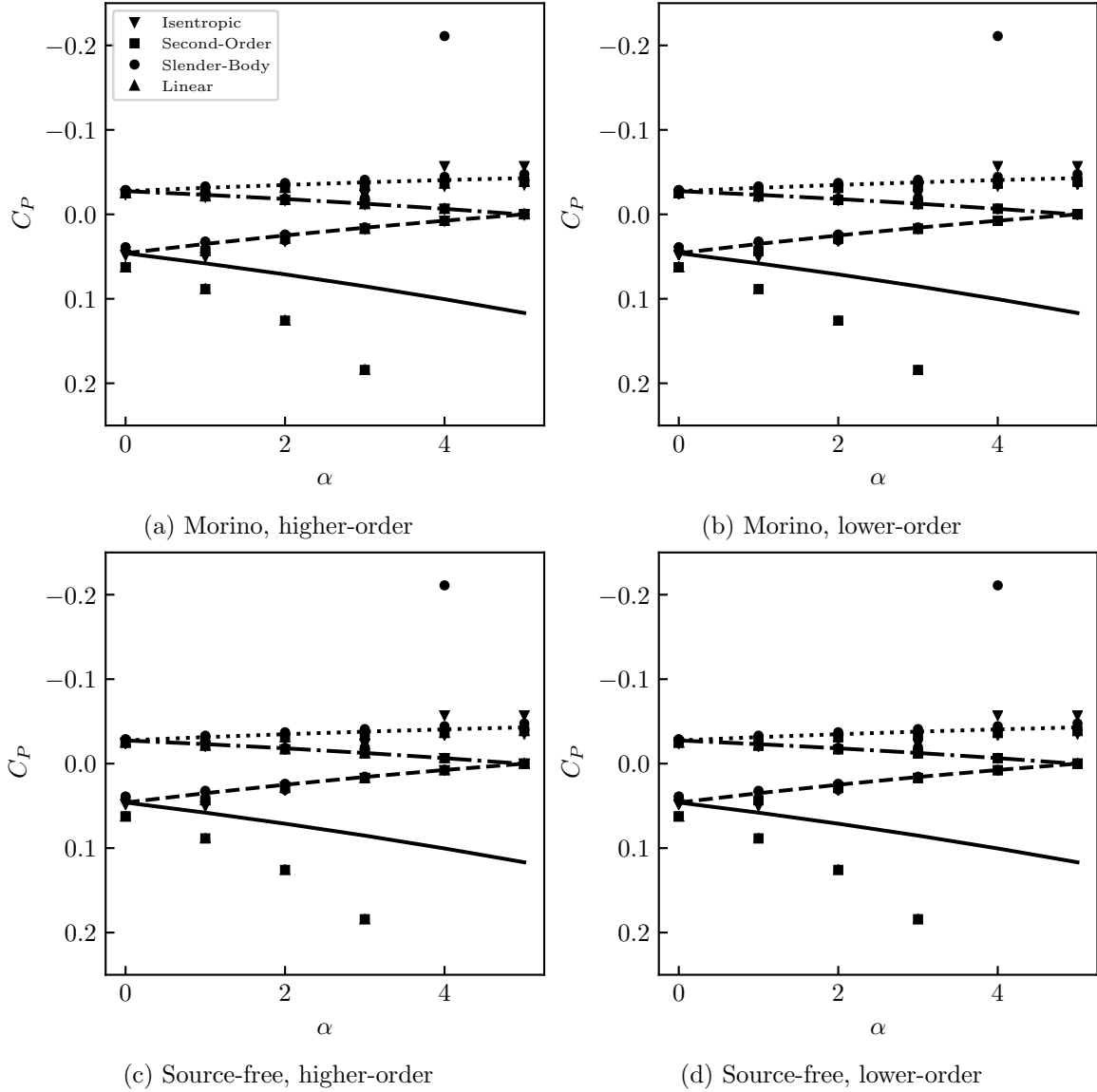


Fig. 9.13: Surface pressures for a 5° half-angle double-wedge airfoil over angle of attack at $M_\infty = 5$ predicted by MachLine (discrete points) and shock-expansion theory (lines). The line style indicates which surface the pressure coefficient is over (see Fig. 9.9). Solid: lower-fore. Dashed: upper-fore. Dash-dot: lower-aft. Dotted: upper-aft.

This makes sense with the assumptions built into a linear panel method. Recall that for a real flow, the entropy of the flow increases as it passes through a shock wave. However, flow through a Prandtl-Meyer expansion fan is isentropic [141]. Within a panel method based on the Prandtl-Glauert equation, both shocks and expansion fans appear as infinitely weak shocks (the expansion fans being represented by expansive shocks, which are non-physical). The predicted flow though both of these is isentropic within a panel method. Thus, flows aft of a strong shock would not be modeled well by a linear panel method, as the isentropic assumption is not valid. However, flows aft of a weak shock that subsequently encounter a strong expansion could still be modeled well by a linear panel method, since the weak shock is nearly isentropic and the expansion is isentropic, no matter how strong it is. This is what occurs over the upper surface of a double-wedge airfoil at finite angles of attack. As the angle of attack increases, the upper bow shock becomes weaker and more-nearly isentropic. The expansion fan that transitions the flow from the forward to the aft surface is always isentropic. Hence, at higher angles of attack, MachLine was able to accurately predict the pressure coefficient on the upper-aft surface despite its large inclination relative to the freestream.

Interestingly, the pressure predictions from MachLine for the lower-aft surface remained accurate even up to high Mach numbers and angles of attack. At this point, the flow had passed through a strong shock and a subsequent expansion fan. Thus, the assumption of isentropic flow built into MachLine was not valid. Despite this, the flow prediction was accurate. The reason for this unexpected accuracy is not known but likely has to do with the small inclination of the lower-aft surface relative to the freestream (i.e. this surface presented a small perturbation to the freestream).

The total run times averaged across all freestream conditions are listed in Table 9.3. In general, the lower-order singularity distributions resulted in faster run times than the higher-order singularity distributions. The relative performance of the two boundary condition formulations was mixed, with the source-free formulation running faster for the coarser meshes and the Morino formulation running faster for the finer meshes.

Table 9.3: Total run times in seconds averaged across all Mach numbers and angles of attack for the double-wedge wing.

	ML	MH	SL	SH
Coarse	0.094	0.100	0.090	0.101
Medium	0.503	0.606	0.500	0.602
Fine	4.897	5.382	4.985	5.391
Ultra Fine	58.446	64.090	77.766	87.675

9.4 Supersonic Cone Flow

For inviscid, supersonic flow over a cone with a constant half angle and no sideslip or angle of attack, the pressure coefficient on the surface of the cone is constant and may be found from the Taylor-MacColl equations [141]. This fact allows for evaluating the accuracy of MachLine based on two criteria. The first criterion is the accuracy of the pressure results from MachLine compared to the analytic solution. The second criterion is the variation in the surface pressure coefficient predicted by MachLine, which should be minimal. In making the first comparison, it must be remembered that the Taylor-MacColl equations include no linearizing assumptions and so produce exact results for inviscid flows, whereas the results from MachLine will only be approximate.

For this study, results were obtained from MachLine for four different cones having half angles of 2.5, 5, 10, and 15°. For all cones, the pressure coefficient was evaluated using all available pressure rules in order to evaluate relative accuracy. Since the cone was at zero angle of attack and sideslip, only a half mesh was given to MachLine initially, and mirroring was specified in the input. The maximum continuity angle was set to 45°.

An initial study into grid convergence used three levels of mesh refinement for each cone angle. For these meshes, the panel density in both the circumferential and axial directions were refined simultaneously; the number of panels in each direction was doubled between each successive level of mesh refinement. When the lower-order singularity distributions were used, the variations between the results obtained using different mesh refinement levels were minimal for both boundary condition formulations. In addition, the variation in surface pressure coefficient over the cone was negligible when lower-order singularity distributions

were used. However, this was not the case when the higher-order singularity distributions were used. In this case, and for both boundary condition formulations, MachLine predicted significant variations in the surface pressure coefficient between the panels at the tip of the cone, as shown in Figs. 9.14 and 9.15. These variations only appeared near and at the tip. The rest of the cone showed minimal variations in pressure coefficient. The magnitude of the variations also increased with increasing mesh density.

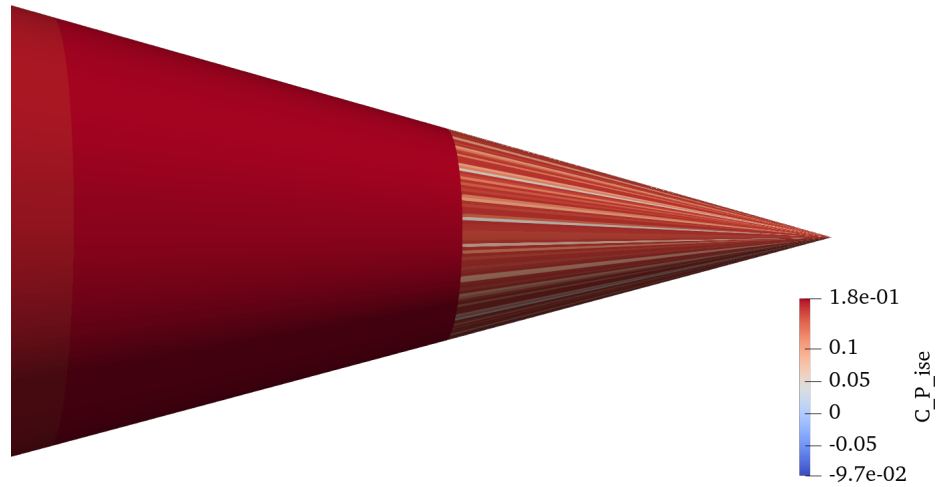


Fig. 9.14: Tip region on the 15° half-angle cone at $M_\infty = 2$ showing the isentropic pressure distribution calculated on the fine mesh when using the Morino formulation and higher-order singularity distributions.

Since the variations increased with mesh density, it was surmised that the variations may have been due to the extremely high aspect ratio of the panels at the tip of the most-refined mesh. This hypothesis was also supported by the fact that the variations were worse for the narrower cones. The cones all had the same number of panels in the circumferential direction, meaning the panels on the narrower cones automatically had higher aspect ratios.

To test this hypothesis, a second set of meshes was generated. For this set, only the axial panel density was varied between the levels of mesh refinement for each cone. In addition, the number of circumferential panels was set to be proportional to the cone angle so that the narrower cones would have fewer panels distributed circumferentially. Thus, the panel aspect ratio remained smaller and more uniform between the various cone angles.

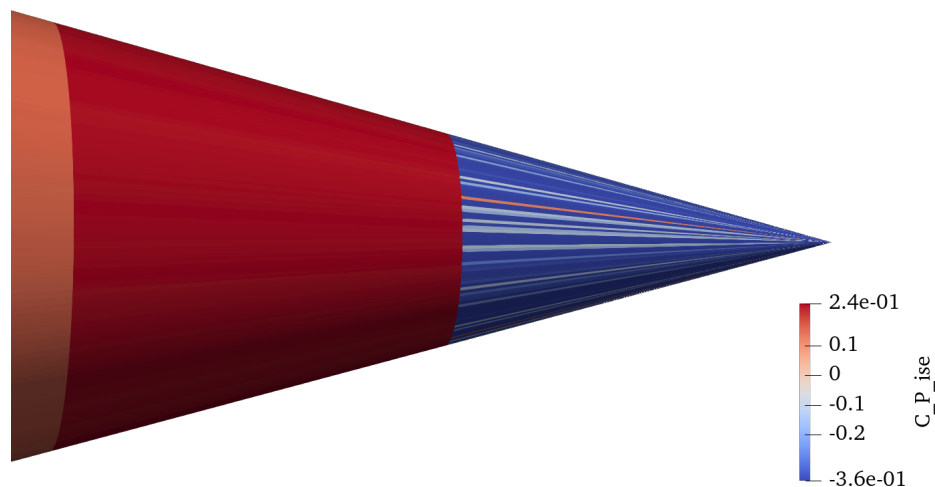


Fig. 9.15: Tip region on the 15° half-angle cone at $M_\infty = 2$ showing the isentropic pressure distribution calculated on the fine mesh when using the source-free formulation and higher-order singularity distributions.

With this new set of meshes, as before, the lower-order singularity distributions resulted in minimal variation in surface pressure coefficient between mesh refinement levels and across the surface of the cone for both boundary condition formulations.

However, even on these improved meshes, the results using higher-order distributions were still poor. In particular, one panel near the tip of each mesh consistently showed a significant deviation in pressure. Regardless of the mesh density, this was the panel lying on the mirror plane on the second row of panels from the tip that had its short edge towards the tip of the cone, as shown in Fig. 9.16. When the higher-order singularity distributions were used, this panel consistently had an unrealistically low pressure coefficient.

In addition to this panel, the panel opposite it on the same row of panels typically showed erroneous results. Both of these panels abutted the mirror plane, meaning there was a qualitative shift in the arrangement of the mesh next to these panels. This shift in arrangement is shown in Fig. 9.17. At the mirror plane (the line between the light and dark gray panels), the orientation of the two triangular panels making up a quad switched. Otherwise, the orientation of this diagonal division between panels remained constant around the cone. This same type of behavior was seen on the leading edge of the ONERA wing (see Section 9.2, particularly the discussion regarding Fig. 9.5). Clearly, the midpoint-

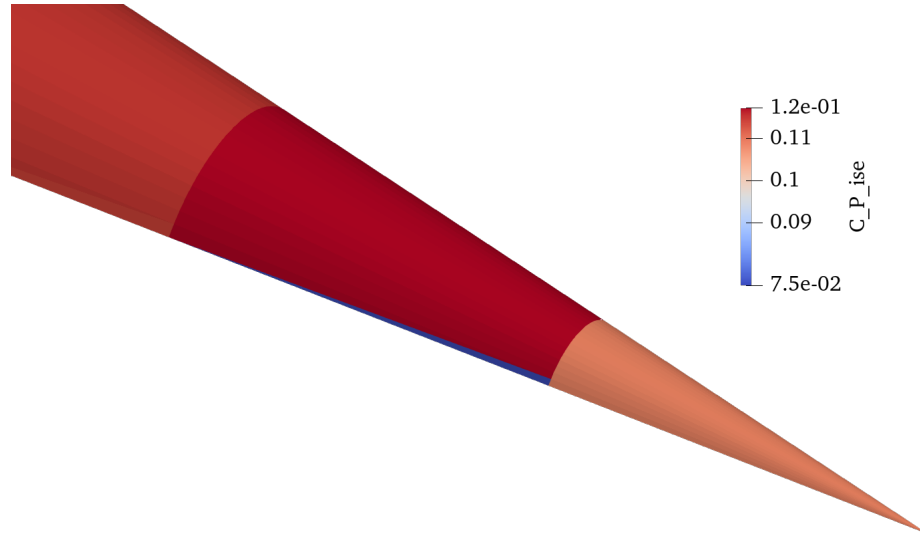


Fig. 9.16: Rendering of the isentropic pressure distribution at the tip of the 10° cone at $M_\infty = 1.7$ using Morino formulation and higher-order distributions on the fine mesh. The erroneous panel is the blue one (extremely low pressure) at the bottom of the second row from the tip.

interpolation method is sensitive to such qualitative shifts in the mesh. Downstream of the erroneous panels just discussed, there were also slight oscillations in the surface pressure coefficient. These oscillations appeared to be induced by the panels just discussed. Besides these erroneous panels on the mirror plane and the pressure variations they induced downstream, the results obtained using the higher-order distributions were much improved by keeping the mesh density in the circumferential direction constant. Even at the highest mesh density, the large variations in pressure coefficient seen in Figs. 9.14 and 9.15 were not present.

To test the dependence of these issues on the sudden qualitative shift in the mesh at the mirror plane, a third set of meshes were generated that this time modeled the entire cone, not just one half. With these meshes, mirroring was not needed, and so the sudden change in panel arrangement did not appear. When the full meshes were used, the results obtained using the higher-order singularity distributions were significantly improved. The results still showed some oscillations in the pressure coefficient near the tip. It should be kept in mind that even going from the first row of panels at the tip of the cone to the

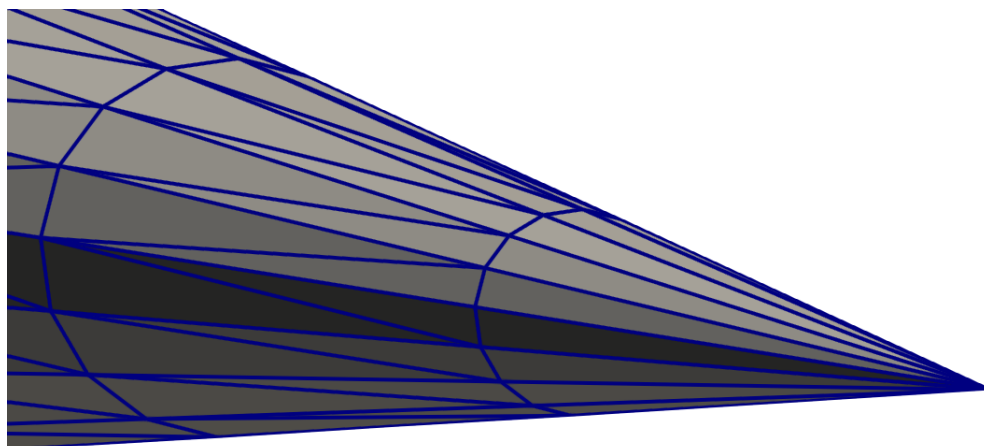


Fig. 9.17: First and second rows of panels at the tip of the cone showing both the original (light gray) and mirrored (dark gray) halves of the mesh.

second row of panels represents a qualitative shift in the mesh; the first row is made up of triangles having all different normal vectors, whereas the second row of panels consists of quads divided into triangles with each pair having identical normal vectors. Unfortunately, this aspect of the meshing could not be avoided. The oscillations caused by this were not large for most cases, and so the third set of meshes (those with constant refinement in the circumferential direction and modeling the full cone) was deemed satisfactory. For all subsequent comparisons, the most-refined mesh from this third set was used.

The solutions to the Taylor-MacColl equations used here were taken from [5]. The surface pressure predictions from MachLine over a range of Mach numbers for the four cones compared to the Taylor-MacColl results are shown in Figs. 9.18, 9.21, 9.22, and 9.23. These plots show the average pressure coefficient over the surface of the cones with uncertainty bands denoting the standard deviation of the pressure coefficient.

At the smallest cone angle of 2.5° (Fig. 9.18), the predictions from MachLine using the lower-order singularity distributions were very good up to the highest Mach number considered. The predicted pressures were close to the analytic result and there was minimal variation in the pressure coefficient across the cone surface.

When the Morino formulation was used with higher-order singularity distributions, there were variations in pressure coefficient across the cone surface that grew in magnitude

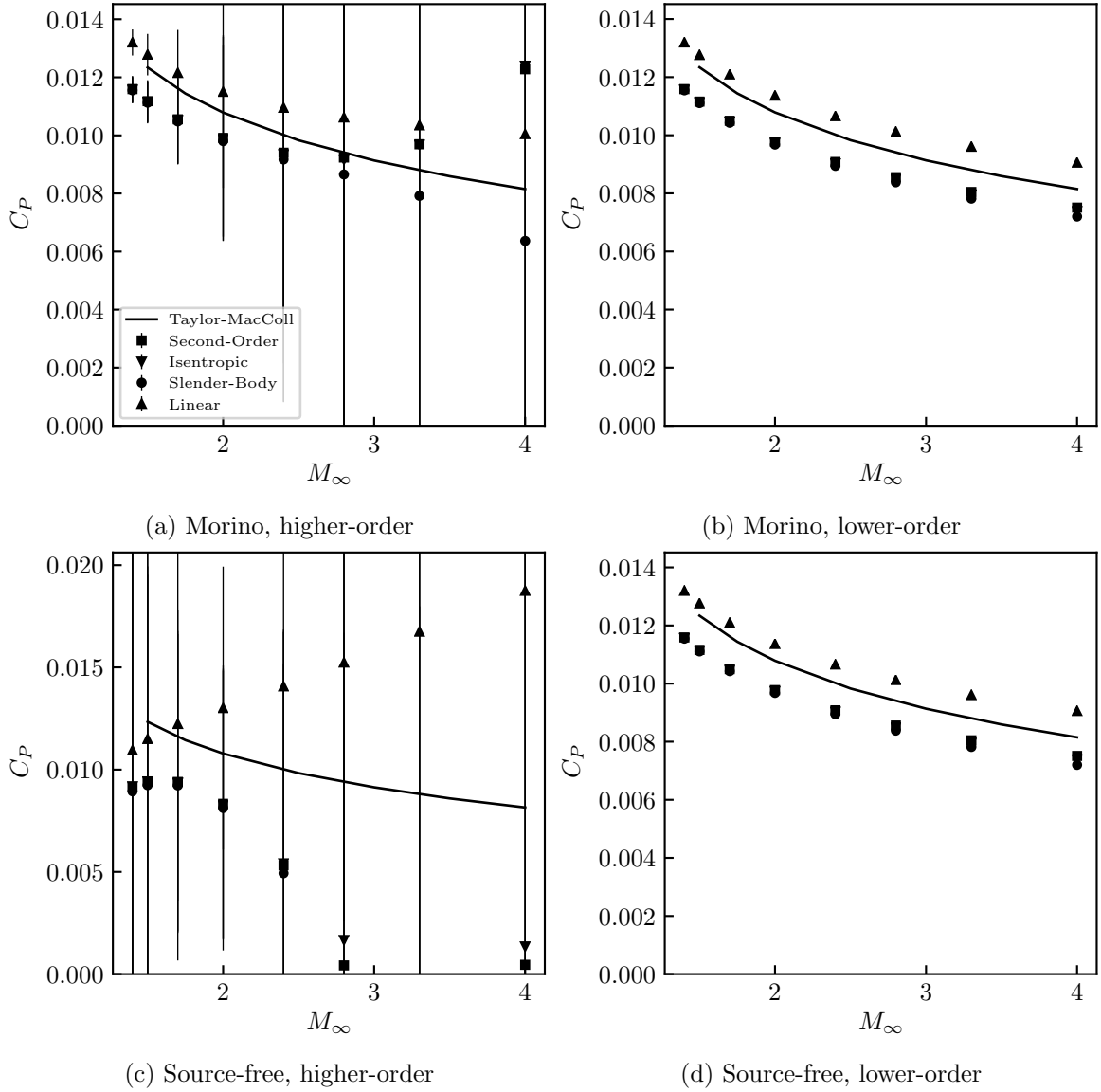


Fig. 9.18: Average surface pressure coefficients for supersonic flow over a 2.5° half-angle cone as a function of Mach number compared to results from the Taylor-MacColl equation [5]. Error bars represent the standard deviation in the pressure coefficient over the whole surface of the cone.

with increasing Mach number. Examining the results at the highest Mach number revealed that the panels touching the trailing edge had an abnormally high pressure coefficient. In addition to increasing with Mach number, this high pressure coefficient increased with mesh density, as shown in Fig. 9.19. As with the results in Section 9.2, this was likely due to how discontinuous edges are treated using the midpoint-interpolation method. For discontinuous

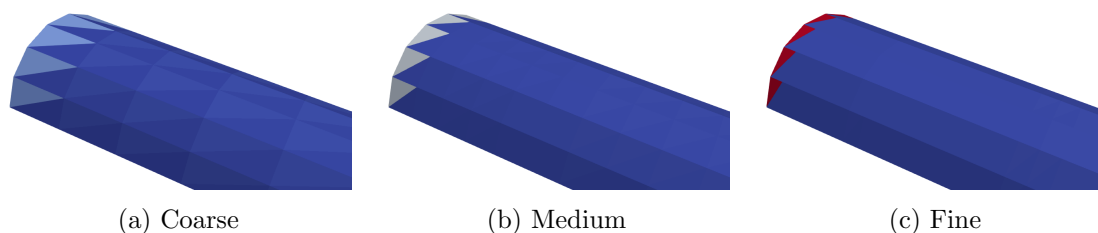


Fig. 9.19: Aft portion of the 2.5° half-angle cone at $M_\infty = 3.3$ showing erroneous pressure results at the trailing edge as a function of mesh density. The color scale ranges from $C_P \approx 0.002$ (dark blue) to $C_P \approx 0.7$ (dark red).

edges (such as these trailing edges), the midpoint strength was set to be the average of the two endpoint strengths, rather than being fit from four neighboring vertices, resulting in the doublet strength distribution along that edge being linear. All other edges parallel to these edges in the mesh (i.e. the edges running circumferentially) were not marked as discontinuous and so had quadratic doublet strength distributions. The shift from quadratic variation to linear variation across the panel was likely what caused the erroneous pressure results. As the mesh was refined, this shift was forced to occur over a shorter distance, amplifying the poor results.

This issue could potentially be fixed by improving how discontinuous edges are treated in the midpoint-interpolation method. This could be particularly improved for edges that abut empty space. Since there is no need to match doublet strength with another panel across such an edge, the linear constraint for discontinuous edges in the midpoint-interpolation method could be relaxed to better model the flow.

When the source-free formulation was used with the higher-order singularity distributions, the results were particularly poor. Only at the lowest Mach numbers were the results from MachLine at all close to the analytic result. In addition, there were consistently large variations in pressure coefficient at all Mach numbers. For example, the isentropic pressure distribution over the 2.5° half-angle cone at $M_\infty = 1.7$ is shown in Fig. 9.20. As can be seen, there were significant variations in the pressure coefficient everywhere on the surface of the cone. This issue likely stemmed from a combination of the high aspect ratio panels at the tip (note the extremely low pressure coefficient on the first row of panels) and the

forced shift to a linear distribution at the back (note the extremely high pressure coefficient on the panels abutting the rear face). However, these poor results were limited to the 2.5° half-angle cone. At higher cone angles, the source-free formulation with higher-order singularity distributions produced better results.

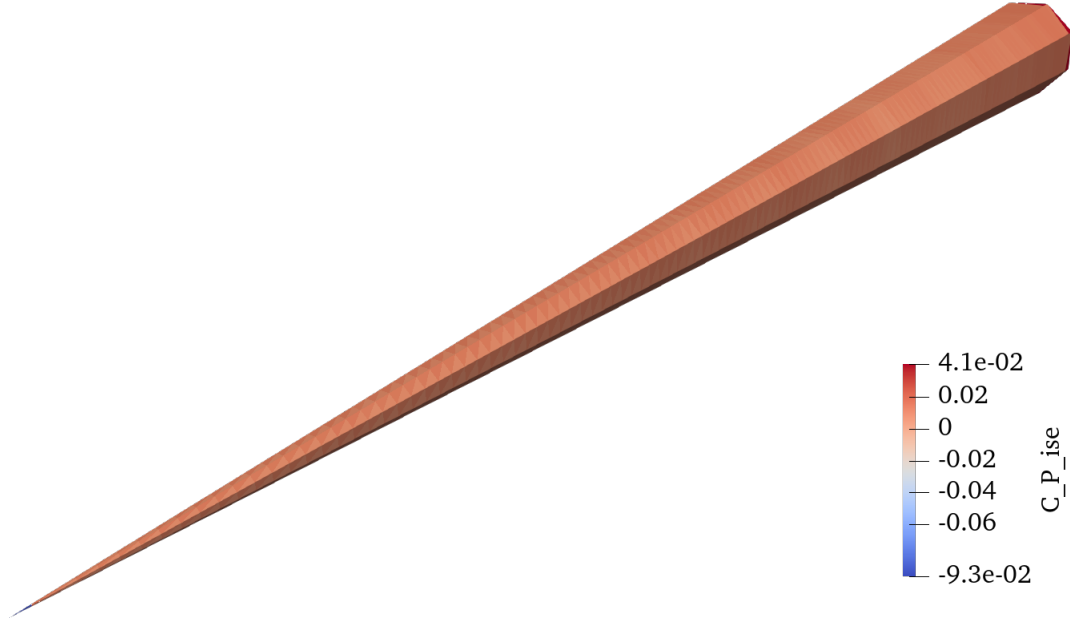


Fig. 9.20: Isentropic pressure distribution over the 2.5° half-angle cone at $M_\infty = 1.7$ using higher-order singularity distributions and the source-free formulation on the finest mesh.

It should be noted that the Morino formulation, even when paired with higher-order singularity distributions, did not produce results like those seen in Fig. 9.20 for the 2.5° half-angle cone. This highlights the improved performance of the Morino formulation over the source-free formulation when higher-order singularity distributions are used.

At a cone angle of 2.5° , the results from MachLine generally followed the analytic result well across the range of Mach numbers considered. However, as the cone angle was increased, the average results from MachLine began to diverge from the analytic solution, beginning at the highest Mach numbers. At a cone angle of 5° (Fig. 9.21), the results from MachLine were still very close to the analytic result at $M_\infty = 4$, though not as close with the 2.5° cone. As before, the lower-order distributions resulted in minimal variations

in pressure coefficient across the surface of the cone. When the higher-order singularity distributions were used, the Morino formulation again resulted in variations that increased with Mach number. And with the higher-order singularity distributions, the source-free formulation resulted in pressure variations at all Mach numbers. However, these variations were not as severe as before, and the average pressure coefficient was still quite accurate.

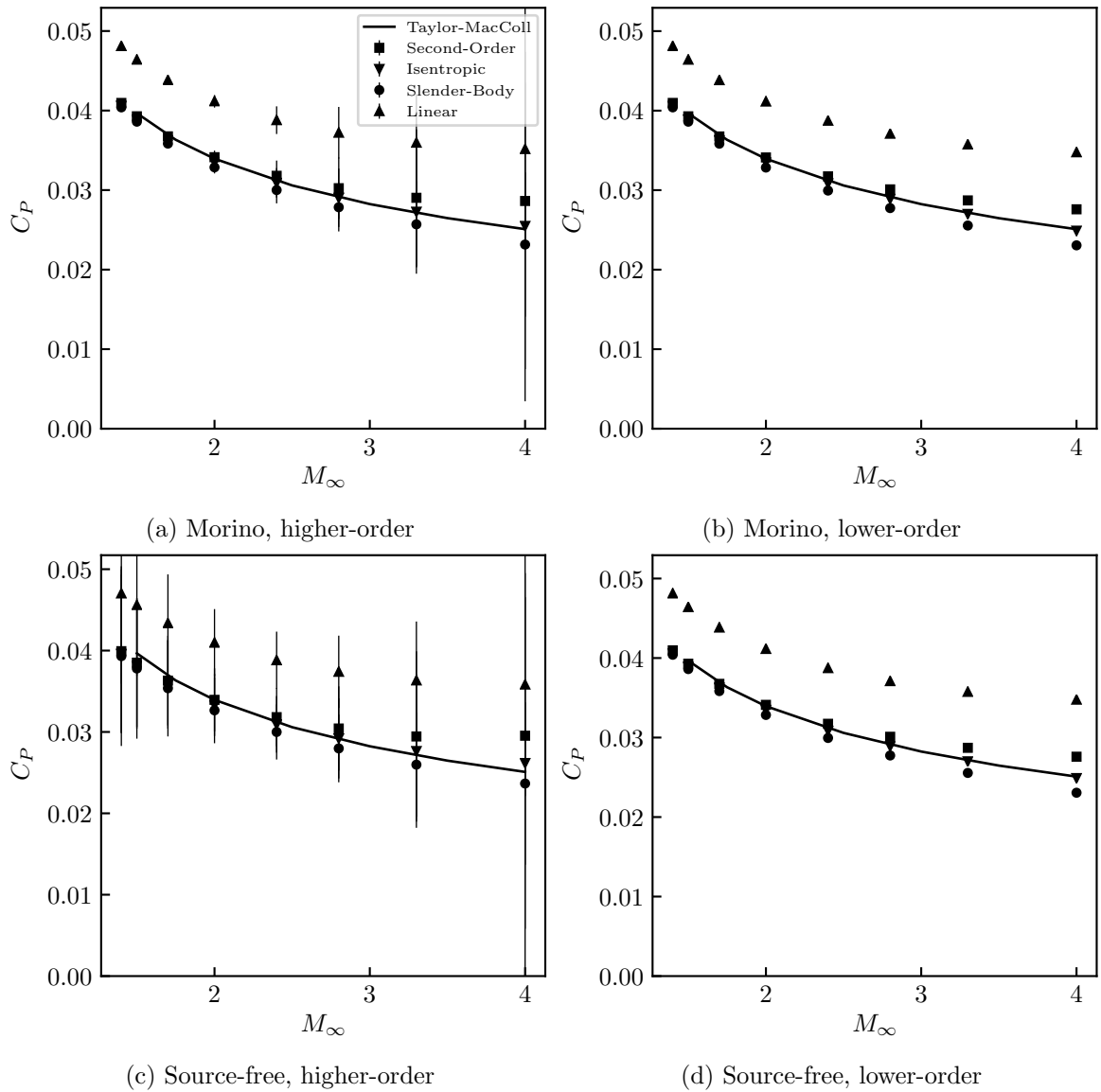


Fig. 9.21: Average surface pressure coefficients for supersonic flow over a 5° half-angle cone as a function of Mach number compared to results from the Taylor-MacColl equation [5]. Error bars represent the standard deviation in the pressure coefficient over the whole surface of the cone.

At a cone angle of 10° (Fig. 9.22), the results from MachLine were only good up to approximately $M_\infty \approx 2.5$. Here, the variations in pressure coefficient due to the higher-order singularity distributions were much less than before, with the Morino formulation still resulting in less variation than the source-free formulation

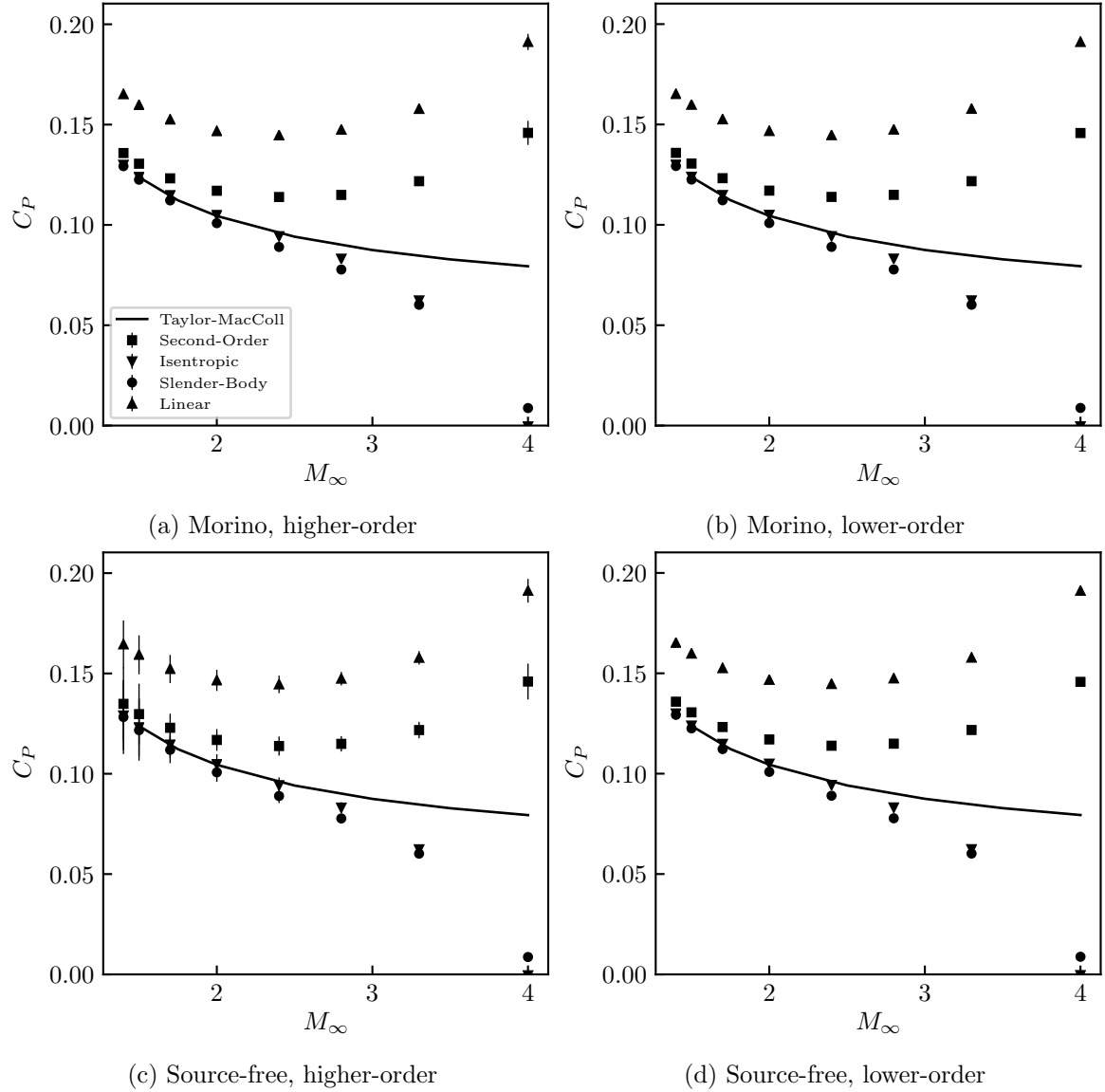


Fig. 9.22: Average surface pressure coefficients for supersonic flow over a 10° half-angle cone as a function of Mach number compared to results from the Taylor-MacColl equation [5]. Error bars represent the standard deviation in the pressure coefficient over the whole surface of the cone.

At a cone angle of 15° (Fig. 9.23), the results from MachLine diverged quickly above $M_\infty \approx 1.75$. here, the only noticeable variations in pressure coefficient appeared when the source-free formulation was used with the higher-order singularity distributions, and even then only at the lowest Mach numbers.

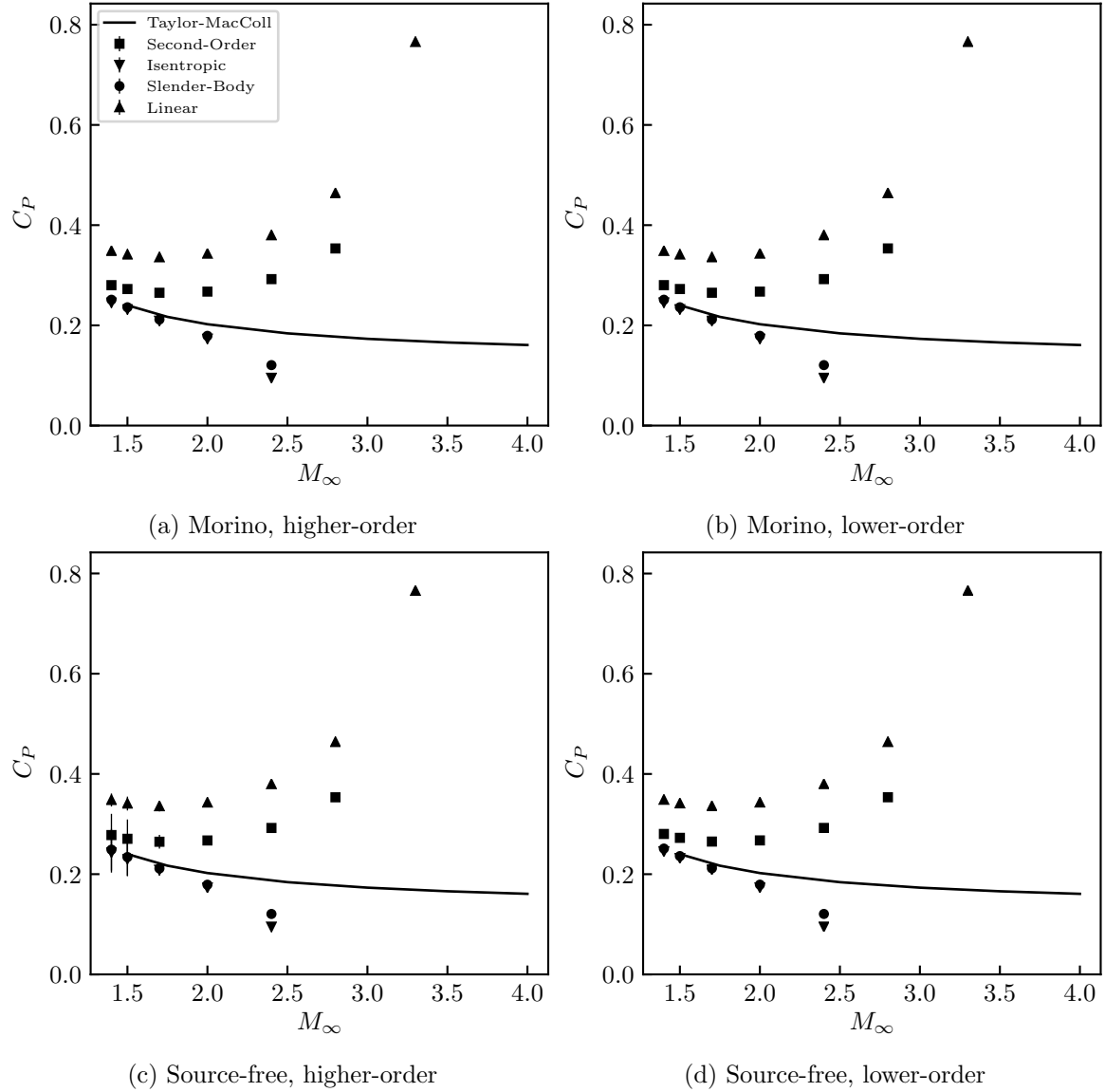


Fig. 9.23: Average surface pressure coefficients for supersonic flow over a 15° half-angle cone as a function of Mach number compared to results from the Taylor-MacColl equation [5]. Error bars represent the standard deviation in the pressure coefficient over the whole surface of the cone.

Overall, the slender-body and isentropic pressure rules tended to track each other and the analytic result most closely. This made sense, as these cones would generally be considered slender. On the other hand, the linear and second-order pressure rules consistently over-predicted the pressure coefficient on the surface of the cone for all cone angles and Mach numbers.

As with the double-wedge wing (Section 9.3), the relative spread of the different pressure rules again gave a good indication of the accuracy of the results. The spread between the different pressure rules correlated well with the error relative to the analytic solution.

For all cone angles, the lower-order distributions resulted in negligible variations in pressure coefficient (the standard deviation in pressure coefficient was on the order of 10^{-7} for all cone angles with the lower-order distributions). In addition, for all cone angles except 2.5° , the Morino formulation paired with the higher-order singularity distributions also resulted in negligible variations in pressure coefficient across the surface of the cone. Only the source-free formulation paired with higher-order singularity distributions consistently resulted in large variations in pressure coefficient; however, these variations decreased with increasing cone angle and Mach number.

The total run times for each cone are given in Tables 9.4-9.7. As seen with previous cases, the run time was consistently increased by using higher-order singularity distributions over lower-order singularity distributions and by using the Morino formulation over the source-free formulation. As before, the differences in run times for the finest mesh were more affected by the choice of singularity distribution. In addition, at the finest mesh resolution, the source-free formulation paired with the higher-order singularity distributions ran significantly faster than the Morino formulation paired with the higher-order singularity distributions. However as shown already, the pressure predictions coming from the source-free formulation paired with the higher-order singularity distributions were particularly poor, and so the lower run times are not necessarily a good thing.

Table 9.4: Total run times in seconds averaged across all Mach numbers for the 2.5° half-angle cone.

	ML	MH	SL	SH
Coarse	0.134	0.153	0.129	0.141
Medium	0.450	0.482	0.446	0.470
Fine	1.690	3.110	1.682	2.349

Table 9.5: Total run times in seconds averaged across all Mach numbers for the 5° half-angle cone.

	ML	MH	SL	SH
Coarse	0.410	0.429	0.404	0.414
Medium	1.433	1.504	1.439	1.493
Fine	5.435	7.934	5.376	6.749

Table 9.6: Total run times in seconds averaged across all Mach numbers for the 10° half-angle cone.

	ML	MH	SL	SH
Coarse	1.169	1.214	1.145	1.199
Medium	4.388	4.635	4.374	4.526
Fine	16.885	20.567	16.788	18.758

Table 9.7: Total run times in seconds averaged across all Mach numbers for the 15° half-angle cone.

	ML	MH	SL	SH
Coarse	2.135	2.260	2.128	2.222
Medium	8.842	9.123	8.826	8.928
Fine	32.555	38.202	33.032	35.265

9.5 Delta Wing

Love [6] performed wind tunnel experiments on several delta wings in supersonic flow. Here, comparison is made between MachLine and Love's experimental results for wing 11 [6]. This wing had a diamond section, with a maximum thickness of 8% chord located at 18% of the chord length. The leading-edge sweep was 45° , and the wingspan was twice the root chord length. Love reported total lift and drag coefficients, as well as pressure distributions at two spanwise locations, over a range of angles of attack. In addition, Love gave an

analytic solution for the pressure distribution over the wing at zero degrees angle of attack based on a linearized supersonic theory presented by Hayes [8].

An example of the mesh used in MachLine is shown in Fig. 9.24. Two special aspects of generating this mesh need to be explained. First, for each strip of panels in the chordwise direction, the number of panels was set to be proportional to the chord length. This meant that the number of panels in each chordwise strip decreased going from the root to the chord. This had the effect of maintaining approximately the same panel size and aspect ratio over the entire wing, avoiding the higher aspect ratio panel issues seen previously with the higher-order singularity distributions.

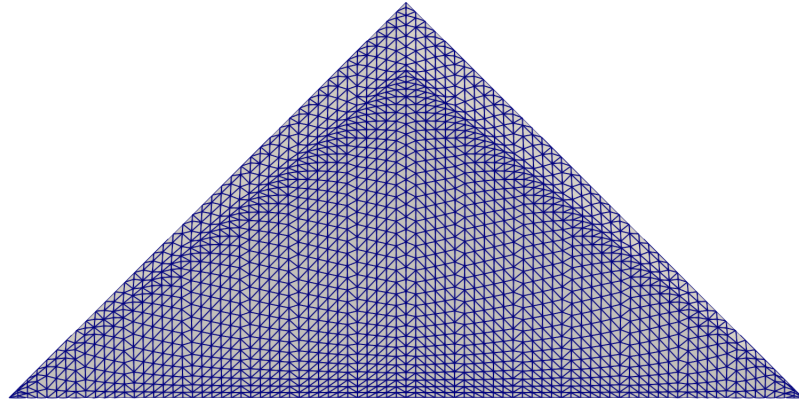


Fig. 9.24: Top-down view of delta wing mesh.

In addition, cosine clustering was used to space the vertices in the chordwise direction between the leading edge and the line of maximum thickness and between the line of maximum thickness and the trailing edge. Rather than using standard cosine clustering (such as described in [144]), the clustering was relaxed somewhat such that the chordwise distribution of vertices was given by

$$x_{\text{vert}_i} = x_{\text{start}} + \frac{1}{2}c \left(1 - 1.08239 \cos\left(\frac{\pi}{8} + i \frac{3\pi}{4(N_{\text{verts}} - 1)}\right) \right), \quad i = 0, 1 \dots N_{\text{verts}} - 1 \quad (9.1)$$

where x_{vert_i} is the location in x of the i -th vertex, x_{start} is the location in x where the distribution is to start, and c is the chordwise extent of the distribution. This resulted in the mild clustering of the panels near the leading edge, the line of maximum thickness, and the trailing edge shown in Fig. 9.24.

The maximum continuity angle for the higher-order distributions was set to 1° . This meant that the leading edges, the line of maximum thickness, and the root line forward of the location of maximum thickness were discontinuous.

Initially, full cosine clustering was used for spacing the vertices in the chordwise direction (as opposed to that given by Eq. (9.1)). Because of this, extremely high aspect ratio panels were created at the leading edge and along the line of maximum thickness. This caused serious issues when the Morino formulation and higher-order singularity distributions were used, as shown in Fig. 9.25. Reducing the level of cosine clustering to what was used for the mesh shown in Fig. 9.24 alleviated this issue. This again demonstrated the sensitivity of the higher-order distributions to high aspect ratio panels.

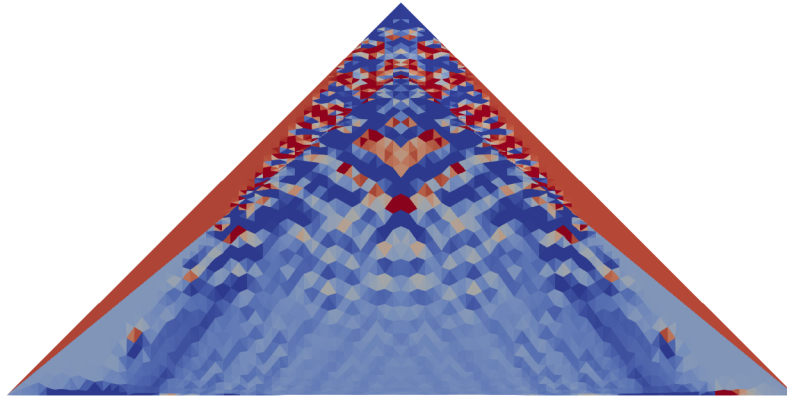


Fig. 9.25: Erroneous isentropic pressure distribution over the top of the Love delta wing at 0° angle of attack using the Morino formulation and higher-order singularity distributions with the heavily cosine-clustered mesh. The color scale ranges from $C_P \approx 1.84$ (dark red) to $C_P \approx -0.54$ (dark blue). The erroneous pressure results were due to extremely fine panels near the leading edge and line of maximum thickness.

The meshes used for this study were the same as used for the grid convergence study in Section 8.2.4. To further establish grid convergence, the lift and drag coefficients were

calculated using MachLine over a range of angles of attack for each mesh. The error in these lift and drag coefficients relative to the most-refined mesh are shown in Figs. 9.26 and 9.27, respectively. For these plots, the marker size decreases with increasing mesh refinement. For all boundary condition formulations and singularity distributions, the grid convergence was very good in C_y . Interestingly, the convergence was much smoother over all angles of attack when the Morino formulation was used, while the source-free formulation resulted in greater variations in grid convergence. The lower-order formulations typically resulted in less relative error in C_y .

The grid convergence results in C_x (Fig. 9.27) showed a stronger dependence upon angle of attack than did the normal force coefficients. The grid convergence was good for all cases. The higher-order singularity distributions resulted in more uniform convergence rates across all angles of attack than did the lower-order singularity distributions (c.f. Fig. 8.16a). Interestingly, the greatest differences were seen here between orders of singularity distributions, rather than boundary condition formulations, as was seen with C_y . Overall, the lower-order singularity distributions resulted in the lowest levels of relative error.

Using the most-refined mesh, which was deemed sufficiently converged, the results obtained from MachLine were compared to the experimental data reported by Love and results from another supersonic panel method, CPanel [145]. The total lift and drag predictions are shown in Figs. 9.28 and 9.29. At all angles of attack, the lift coefficient predicted by MachLine was less than that reported by Love. However, the results from MachLine were closer to the experimental data than the results from CPanel at all angles of attack. This was likely due to a lack of proper wake modeling in CPanel [7]. Among the results obtained from MachLine, there were no noticeable differences between the different boundary condition formulations and singularity distributions.

The drag predictions from MachLine were consistently lower than both the data reported by Love and the predictions from CPanel. It would be expected for MachLine to under-predict the drag compared to wind tunnel data, as MachLine does not account for skin friction. However, it is unclear why the results from CPanel were markedly higher than

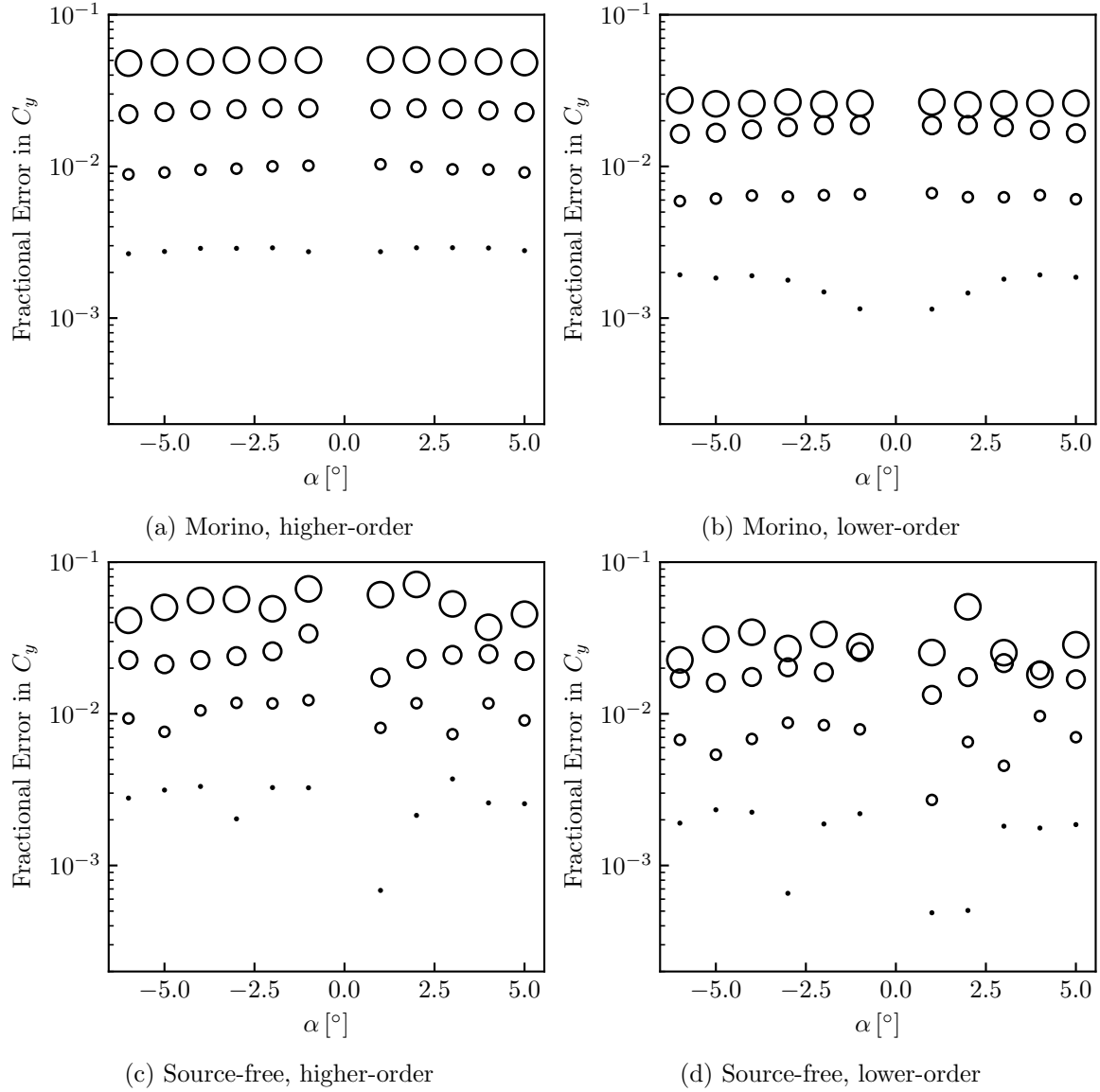


Fig. 9.26: Grid convergence of the normal force coefficient (C_y) for Love's delta wing [6] over a range of angles of attack and $M_\infty = 1.62$.

those from MachLine. This may also have been due to the lack of proper wake modeling in CPanel. CPanel did not allow for a discontinuity in doublet strength at the trailing edge [7], and so the forced continuity may have resulted in a higher drag coefficient.

Figure 9.30 shows the predicted pressure distributions at 0° angle of attack and 22.5% semispan. All combinations of boundary condition formulation and singularity distribution produced excellent predictions compared to the experimental data. In particular, the

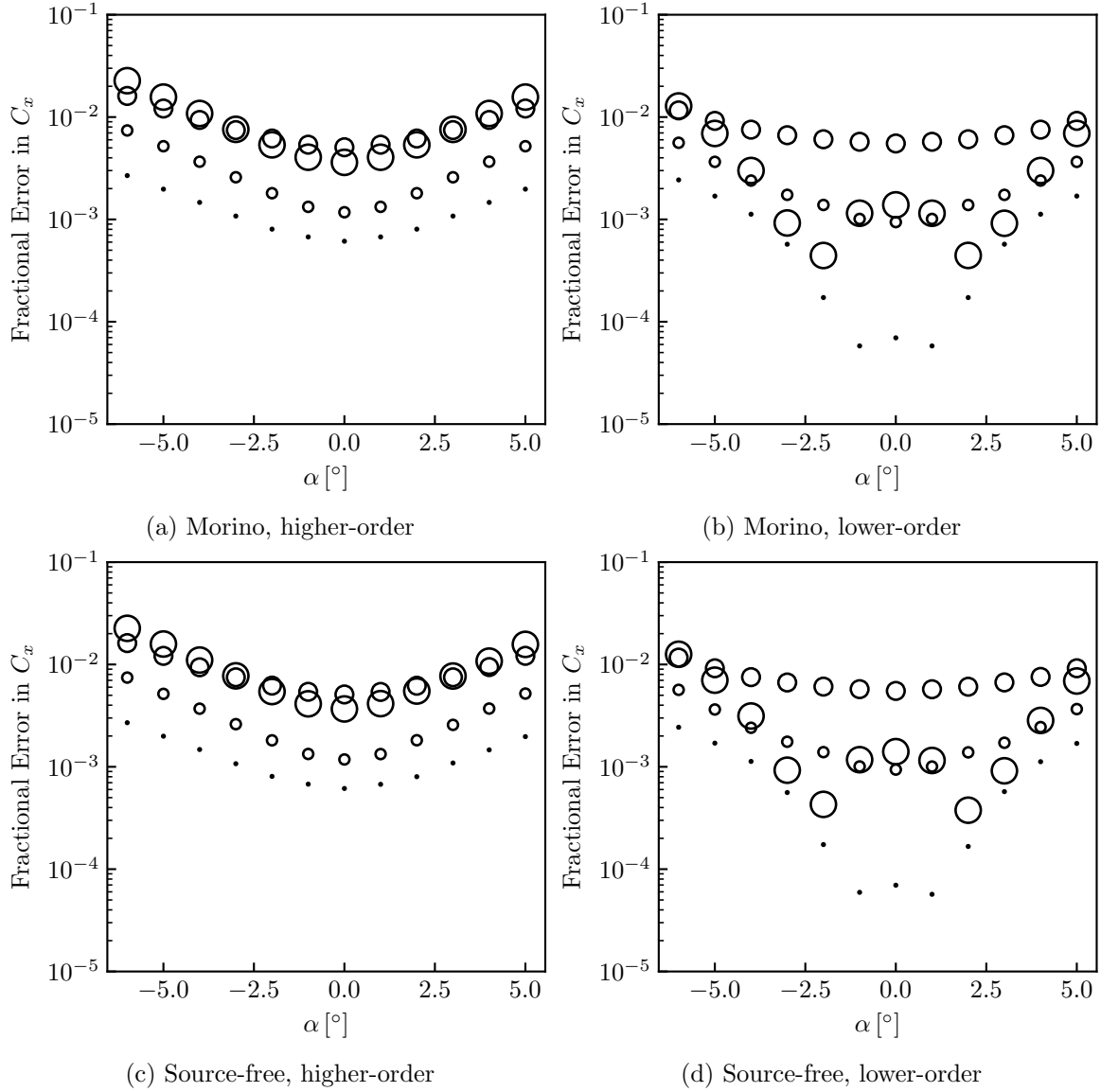


Fig. 9.27: Grid convergence of the axial force coefficient (C_x) for Love's delta wing [6] over a range of angles of attack and $M_\infty = 1.62$.

higher-order singularity distributions resulted in very smooth pressure distributions. The predictions from MachLine were closer to the experimental data than the results from Hayes' linearized solution. While MachLine is based on linearized theory, it does not make linearizing assumptions about the geometry or pressure coefficient, among other aspects. This evidently increases the accuracy of MachLine compared to other linear methods.

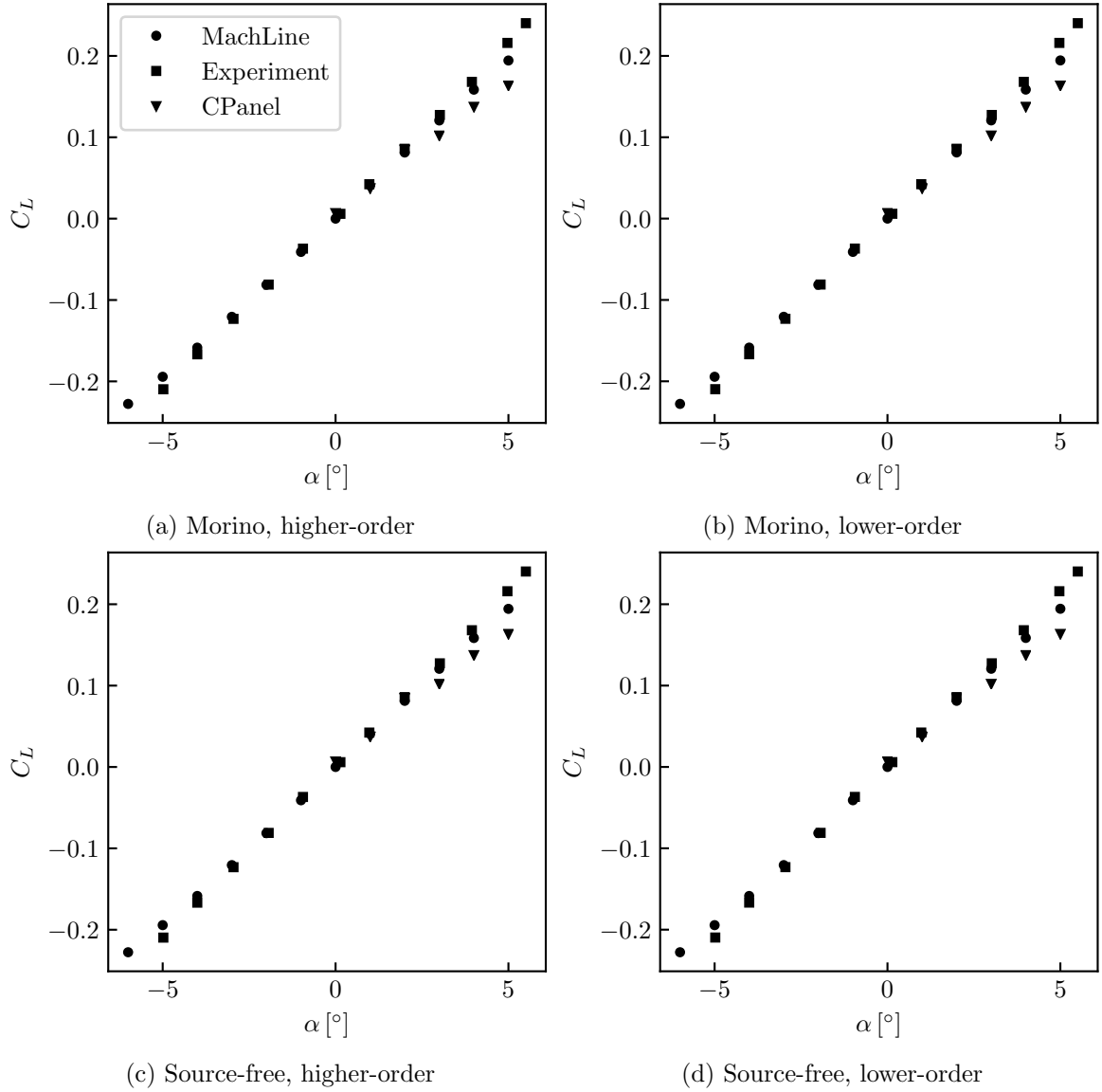


Fig. 9.28: Lift coefficient for Love's delta wing [6] over a range of angles of attack and $M_\infty = 1.62$ from MachLine compared to experimental data reported by Love [6] and predictions obtained from CPanel, as reported by Davis [7].

The results at 64.1% semispan are shown in Fig. 9.31. Here, the predicted pressure distribution from MachLine very closely matched the linearized solution. However, these both did not closely match the experimental data. This was due to the angle of the shock coming off the nose of the wing shown clearly in Fig. 9.32. In linearized theory, the characteristic surfaces (shocks and expansion fans) have a constant angle with respect to

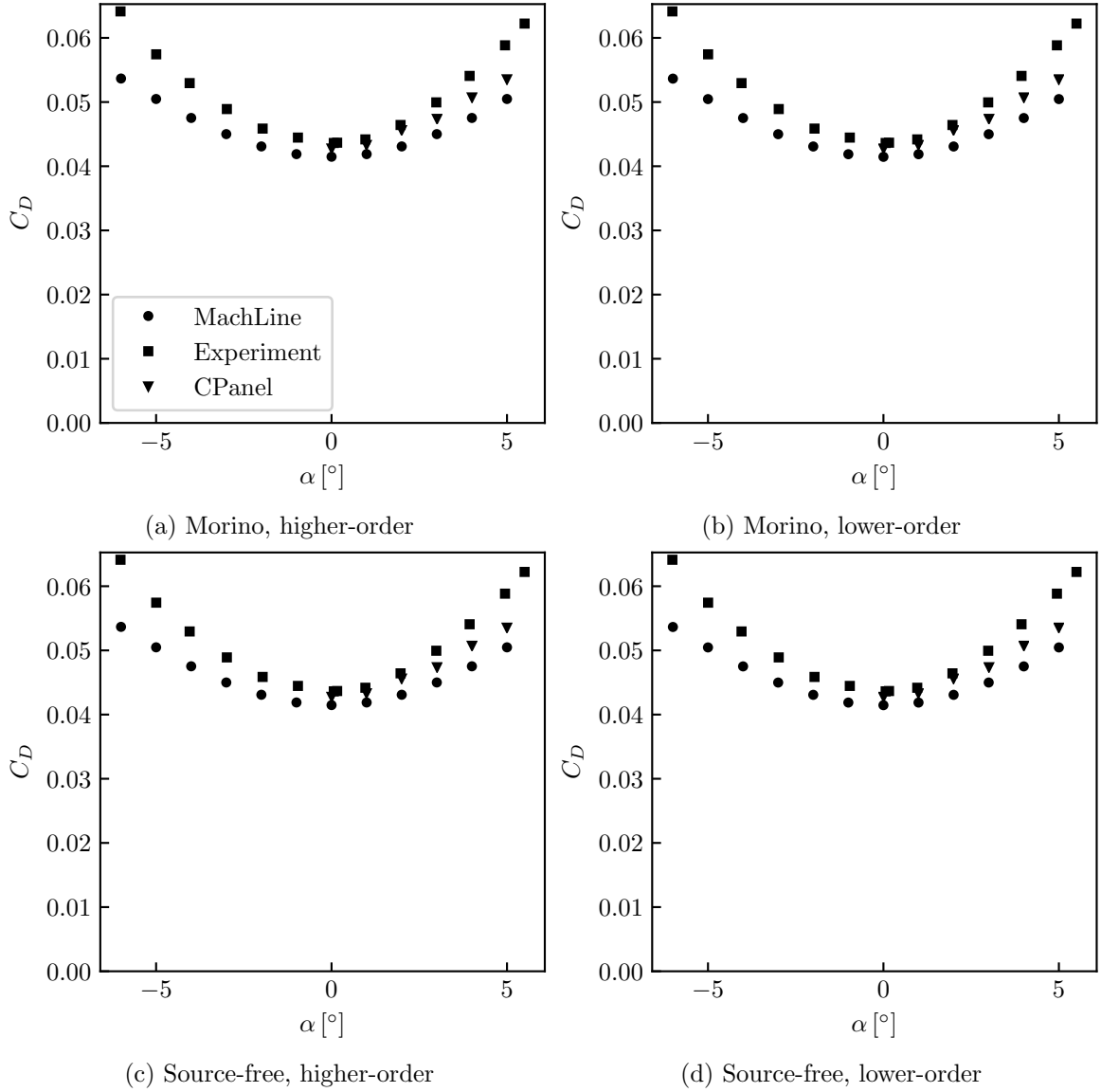


Fig. 9.29: Drag coefficient for Love's delta wing [6] over a range of angles of attack and $M_\infty = 1.62$ from MachLine compared to experimental data reported by Love [6] and predictions obtained from CPanel, as reported by Davis [7].

the freestream. These surfaces also represent the limit of infinitely weak shocks, and so the angle of the characteristic lines in linearized theory are always more shallow than real shocks. Because of this, the experimental data showed a drop in pressure further forward due to the bow shock than either MachLine or the linearized theory predicted. At the trailing edge of the wing, which was aft of both the experimental and linearized characteristics, the

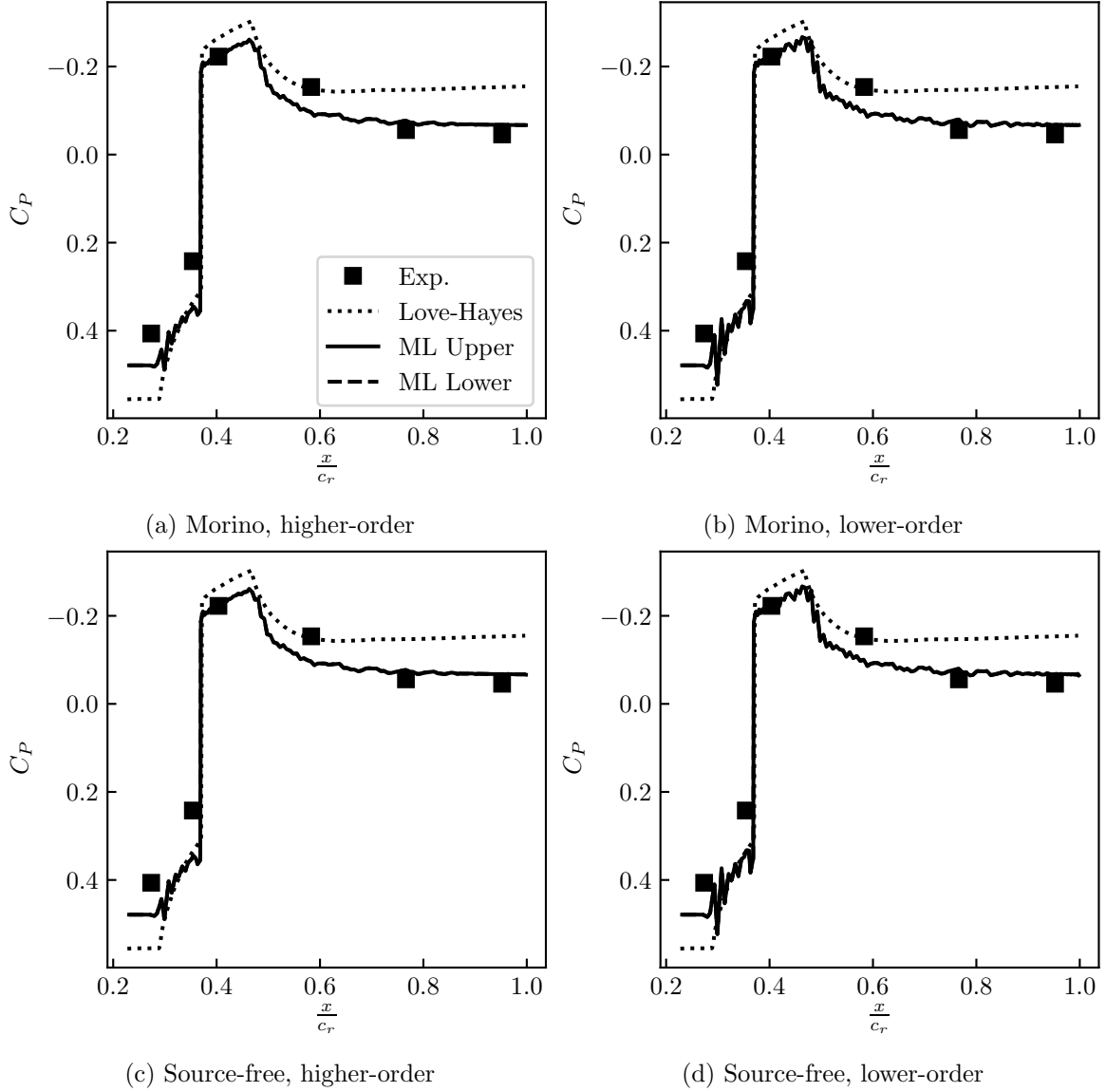


Fig. 9.30: Isentropic pressure distribution at 22.5% semispan and 0° angle of attack from MachLine compared to experimental data reported by Love [6] and an analytic solution based on a linearized theory presented by Hayes [8].

pressure predictions agreed between all three methods. At 22.5% semispan, the slice was close enough to the root to make this difference in characteristic angle insignificant.

As was seen with the pressure distributions at 22.5% semispan, the higher-order singularity distributions resulted in pressure predictions that were much smoother than those obtained from the lower-order distributions.

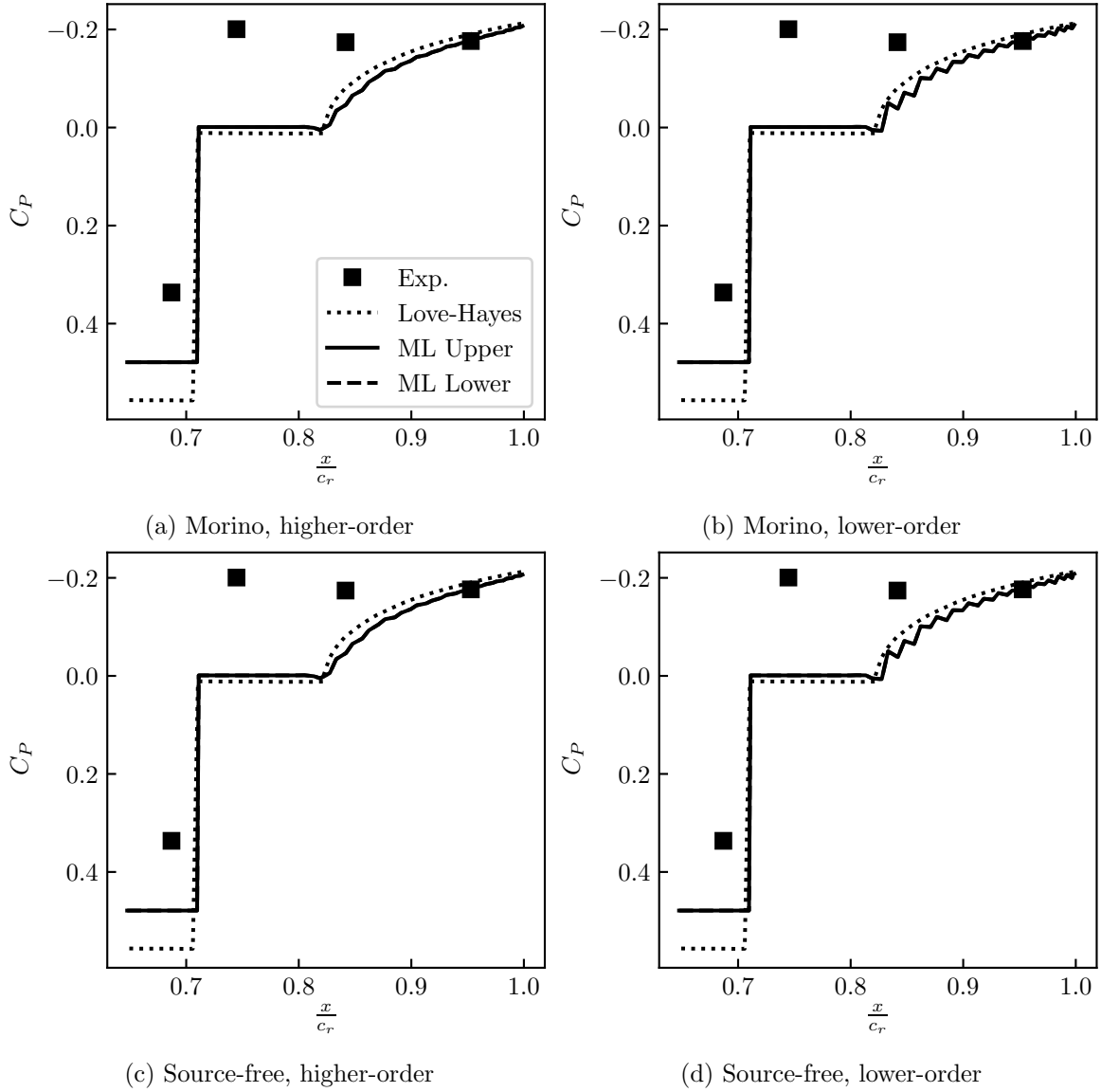


Fig. 9.31: Isentropic pressure distribution at 64.1% semispan and 0° angle of attack from MachLine compared to experimental data reported by Love [6] and an analytic solution based on a linearized theory presented by Hayes [8].

The pressure distributions at 4.1° angle of attack are shown in Figs. 9.33 and 9.34. In general, the predictions from MachLine were closer to the experimental data aft of the location of maximum thickness. The aft portion of the wing had a shallower inclination than the forward portion of the wing, meaning the linearizing assumptions built into MachLine were more valid in that region. As before, the higher-order singularity distributions resulted

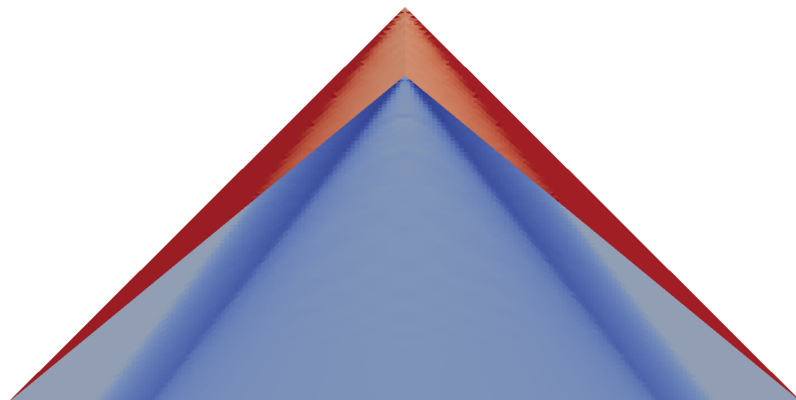


Fig. 9.32: Isentropic pressure distribution over the top of the Love delta wing at 0° angle of attack using the Morino formulation and higher-order singularity distributions showing the characteristic lines (shocks) emanating from the bow and the point of maximum thickness. The color scale indicates high pressures with dark red and low pressures with dark blue.

in smoother pressures than the lower-order singularity distributions, and there were no noticeable differences between the different boundary condition formulations.

As with the results at 0° angle of attack, the difference in characteristic angle between the experimental data and MachLine resulted in a significant difference in pressure predictions at 64.1% semispan. However, at the trailing edge, the pressure predictions matched closely.

The results at 10.75° angle of attack are shown in Figs. 9.35 and 9.36. As at 4.1° angle of attack, the pressure predictions from MachLine at 10.75° were much closer to the experimental data over the aft portion of the wing than over the forward portion. On the forward portion of the wing, the upper surface pressure predictions were reasonably close, while the lower surface pressure predictions were significantly off. This was likely due to the lower surface being steeply inclined to the oncoming flow at this angle of attack, while the upper surface was much less steeply inclined.

Overall, the higher-order singularity distributions produced the most accurate results with no apparent differences between the two boundary condition formulations. The results obtained using the higher-order singularity distributions were smooth and followed the ex-

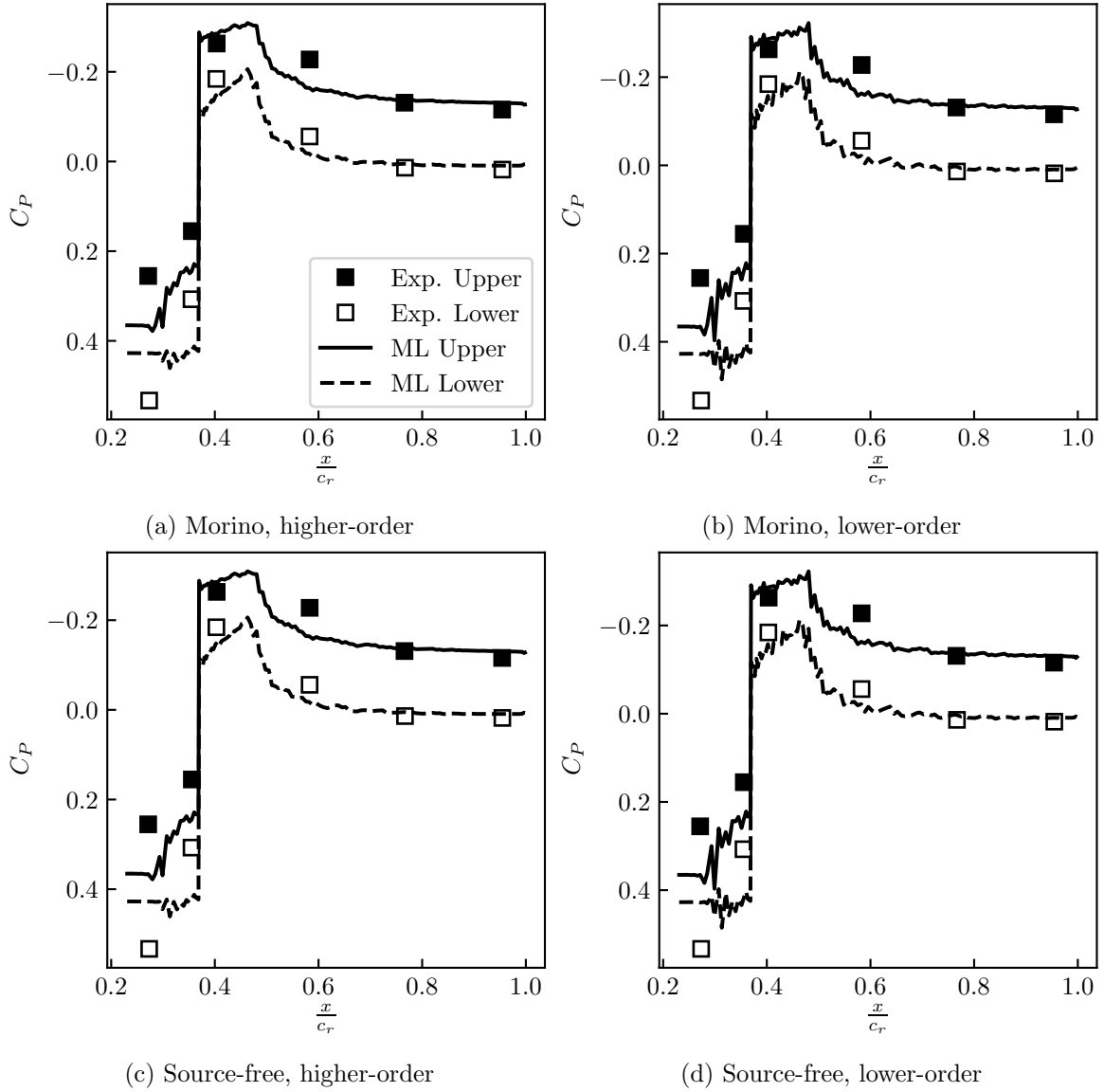


Fig. 9.33: Isentropic pressure distribution at 22.5% semispan and 4.1° angle of attack from MachLine compared to experimental data reported by Love [6].

perimental data well. When lower-order distributions were used, the results from MachLine still followed the experimental data closely but exhibited more numerical noise.

When considering the impact of the different singularity distributions and boundary condition formulations, it is valuable to observe the accuracy of solutions versus the time required to compute them. If for a given mesh, a solution scheme can provide a lower error as fast as or faster than other schemes, then that scheme is superior. Such a comparison

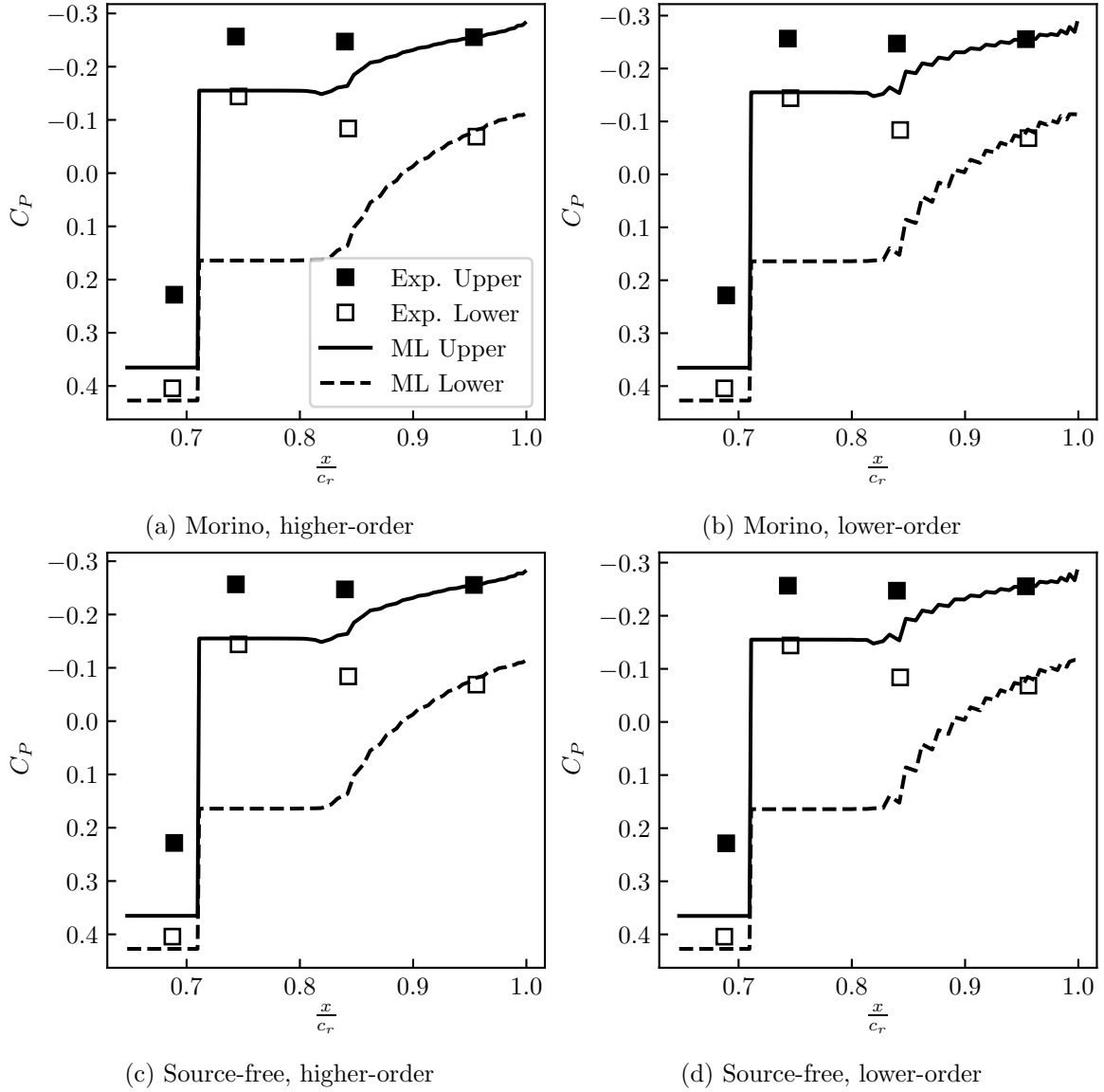


Fig. 9.34: Isentropic pressure distribution at 64.1% semispan and 4.1° angle of attack from MachLine compared to experimental data reported by Love [6].

is made here. Figures 9.37 and 9.38 show the relative error in C_x and C_y , respectively, for Love's delta wing 11 using the different singularity distributions and boundary condition formulations as a function of computation time. For all cases except for C_y at $\alpha = 0^\circ$, the relative error was determined as the fractional error relative to the most-refined result when using the same singularity distribution and boundary condition formulation. For C_y

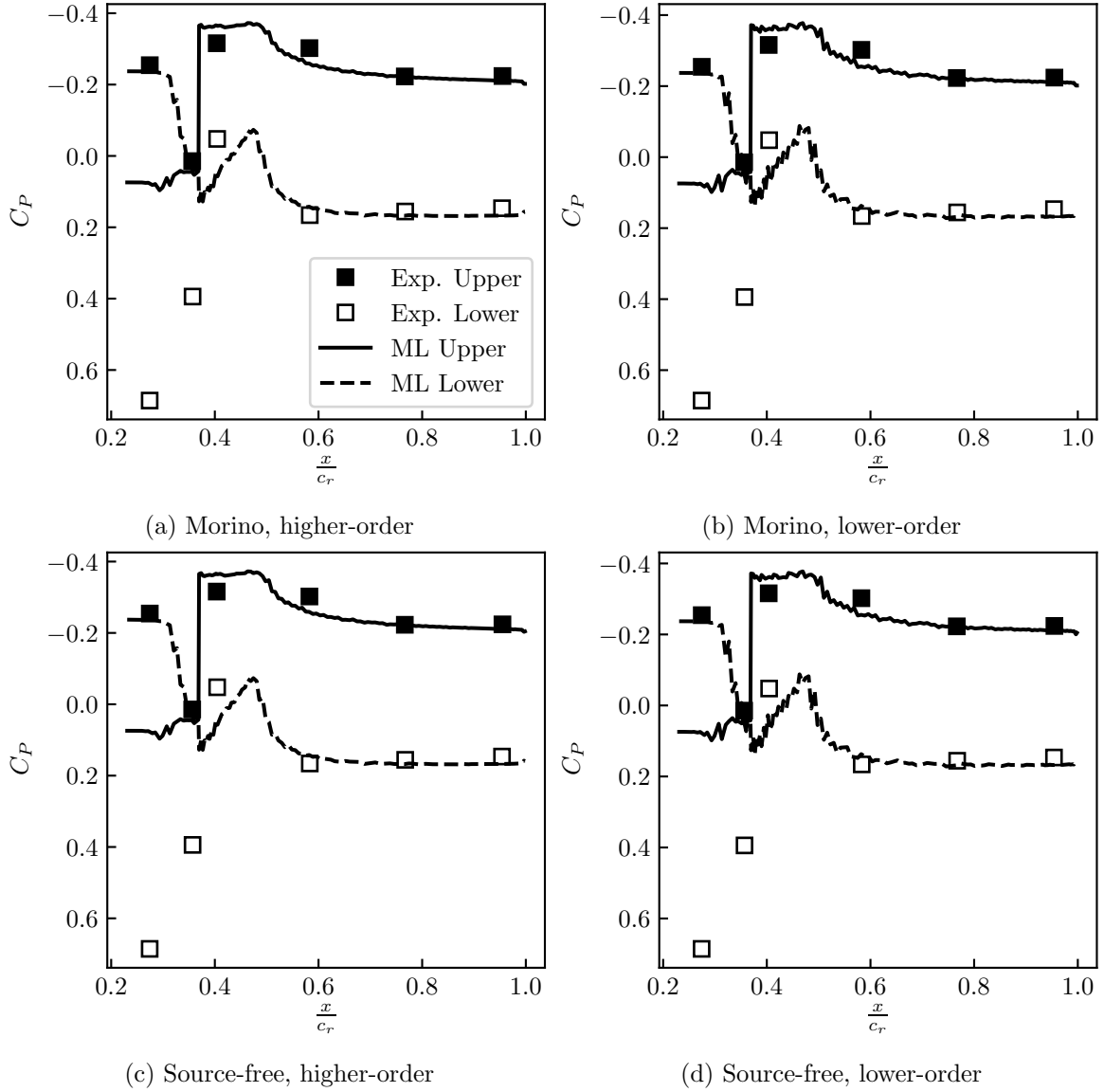


Fig. 9.35: Isentropic pressure distribution at 22.5% semispan and 10.75° angle of attack from MachLine compared to experimental data reported by Love [6].

at $\alpha = 0^\circ$, the relative error was simply the absolute value of C_y , since this should have been zero at $\alpha = 0^\circ$.

For C_x (Fig. 9.37), the lower-order singularity distributions consistently produced lower relative errors than the higher-order distributions for less run time. This was consistent with previous results that have suggested that the higher-order distributions are less accurate and slower than the lower-order distributions. Interestingly, the error in C_x did not mono-

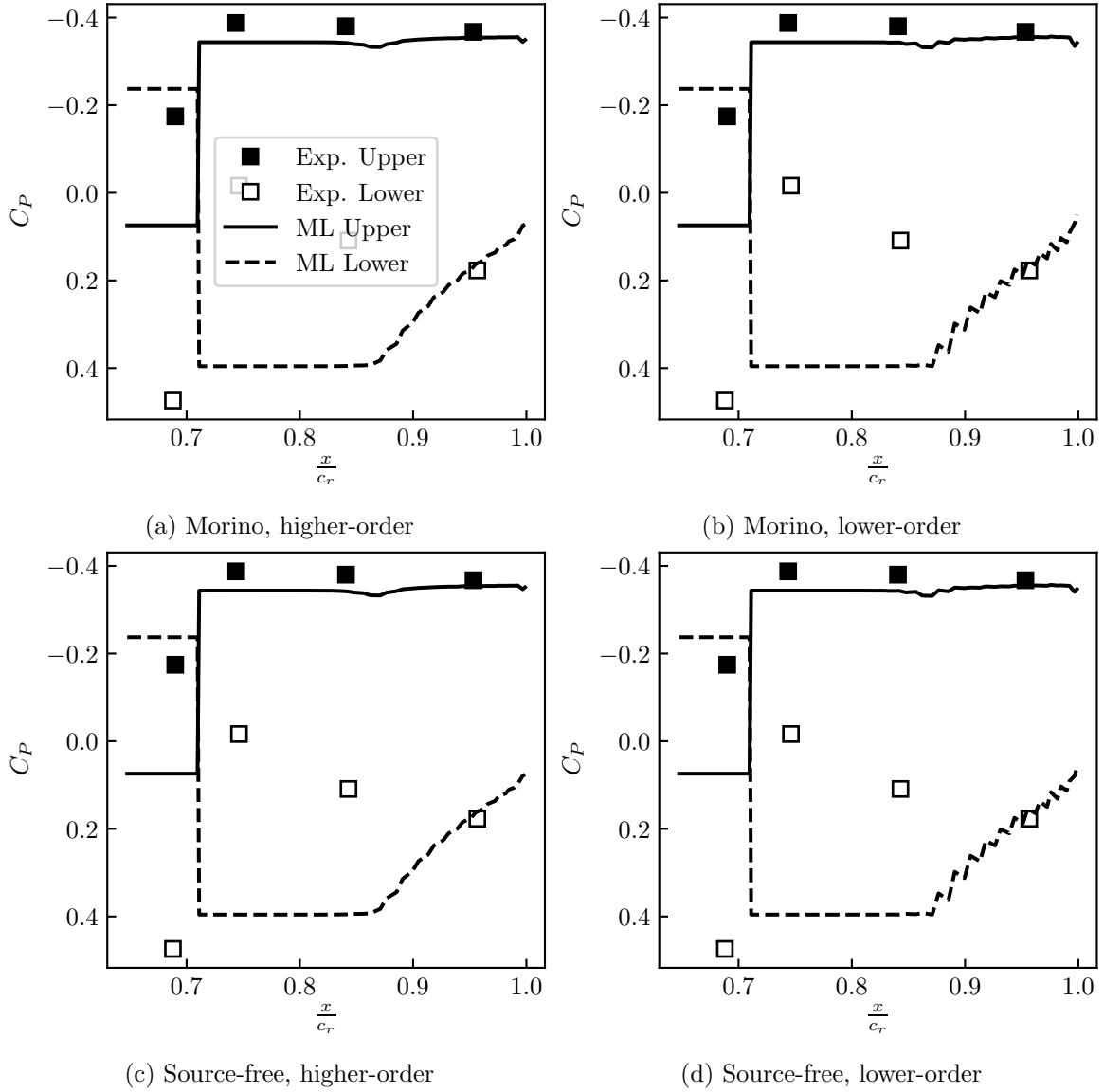


Fig. 9.36: Isentropic pressure distribution at 64.1% semispan and 10.75° angle of attack from MachLine compared to experimental data reported by Love [6].

tonically increase with computation time. Between the two coarsest meshes, there was for all cases a significant increase in error, after which the error decreased monotonically. This could have been why previous estimates of grid convergence were poorly-behaved, as the error in such a case is not well modeled by an exponential function.

For C_y (Fig. 9.38), the lower-order singularity distributions again consistently produced lower relative errors than the higher-order distributions for less run time, though

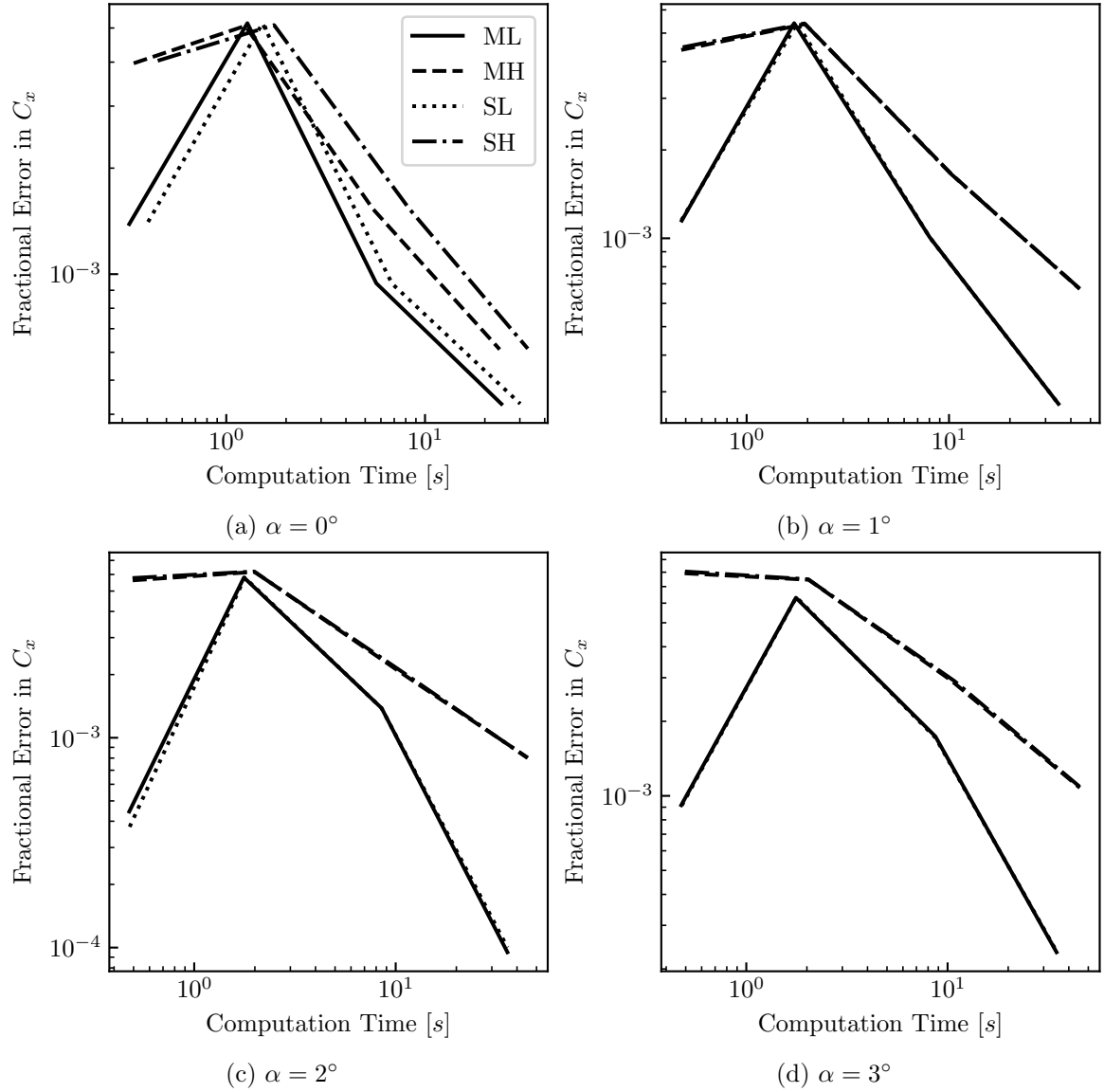


Fig. 9.37: Relative error in C_x on the delta wing for the different boundary conditions and singularity distributions at four different angles of attack as a function of computation time.

the differences between the two orders were less here than they were for C_x . The one exception to this was at $\alpha = 0^\circ$. In this case, the greatest difference appeared between the two boundary condition formulations, rather than the singularity distributions, with the Morino formulation consistently producing lower error than the source-free formulation. These different results were likely due to the way in which the error was calculated at 0°

angle of attack. For all other angles of attack, the fractional error was used, whereas here, the absolute error was used.

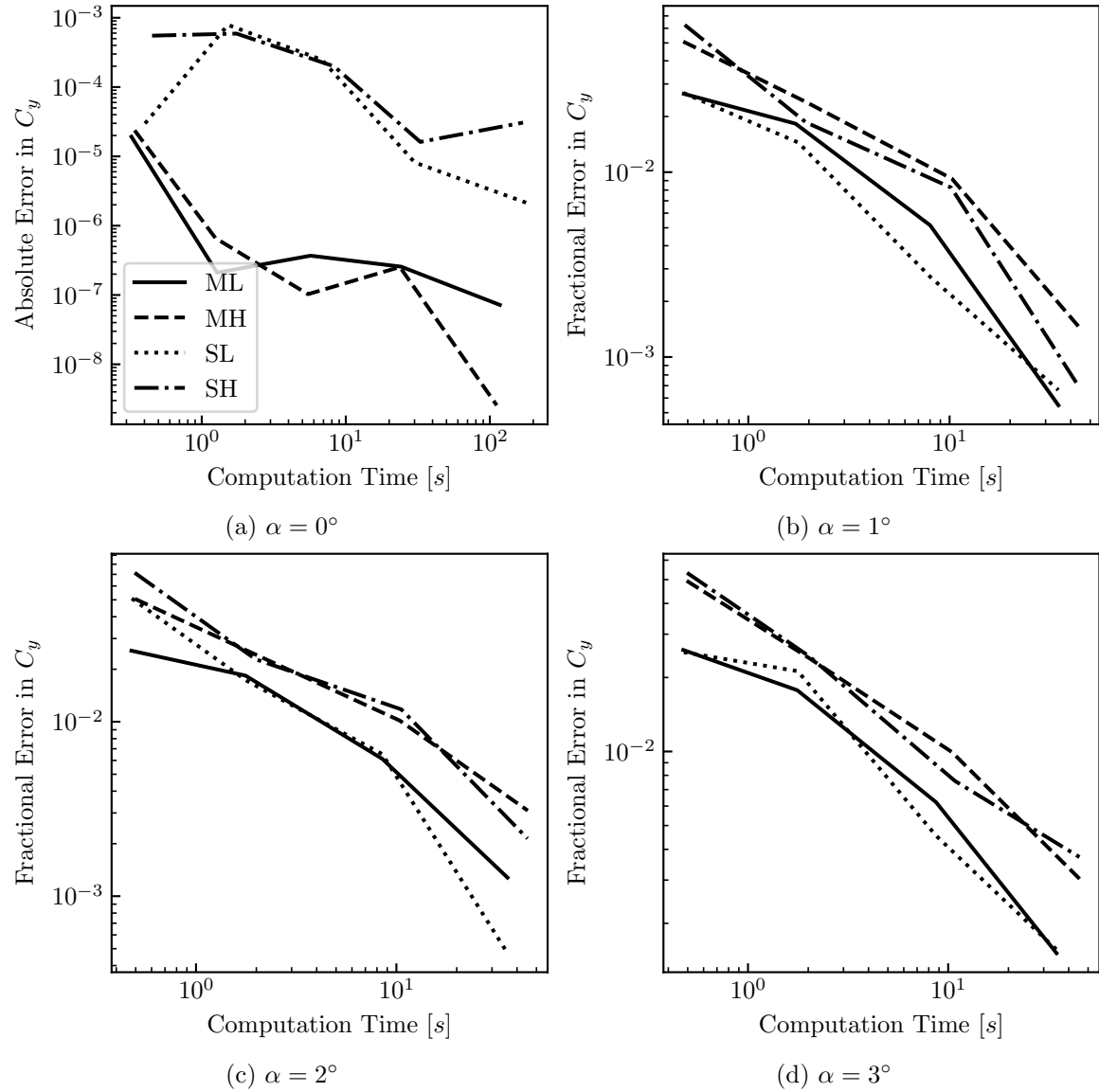


Fig. 9.38: Relative error in C_y on the delta wing for the different boundary conditions and singularity distributions at four different angles of attack as a function of computation time.

9.5.1 Comparison with PAN AIR

PAN AIR has long been used and regarded as the gold standard for supersonic panel methods and has been widely vetted by numerous researchers in numerous situations. As such, it is valuable to compare the performance and accuracy of MachLine to PAN AIR. To compare, the copy of PAN AIR available at [65] was used. The standard Morino formulation was used within PAN AIR. It should also be noted that PAN AIR uses quadratic-doublet-linear-source panels. In addition, at breaks between networks in PAN AIR, only continuity of doublet strength is enforced; continuity of the doublet strength gradient is not enforced. Thus, there is an analog here between the behavior of PAN AIR and the maximum continuity angle used in MachLine.

This particular delta wing geometry presents an opportunity for easily comparing MachLine against PAN AIR. This is because Love's delta wing 11 may be represented in PAN AIR using only four flat networks of body panels (one for each surface facet) and a wake network. An equivalent mesh may also be generated for use in MachLine and the results from the two different codes compared. Here, MachLine and PAN AIR are compared for Love's delta wing 11 in terms of speed, rate of grid convergence, and predicted pressure and forces.

Since PAN AIR operates on structured grids, it uses quadrilateral panels, rather than triangular panels, as MachLine does. Hence, the meshes used by MachLine and PAN AIR cannot be exactly equivalent. To generate the meshes for MachLine and PAN AIR, the same vertices were used. Then for use in MachLine, each quadrilateral panel was divided in two along a diagonal, as shown in Fig. 9.39. Since the results may be sensitive to the choice of diagonal along which this division was made, both directions were tested and the results compared.

For this study, three different levels of mesh refinement were used, with the number of spanwise and chordwise grid points doubling between each successive level of mesh refinement. The delta wing was tested at $M_\infty = 1.62$ and angles of attack ranging from 0° to 5° .

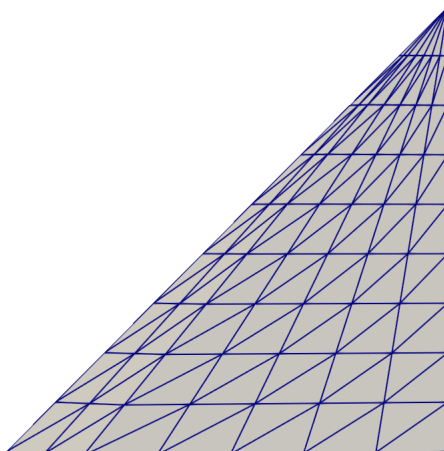
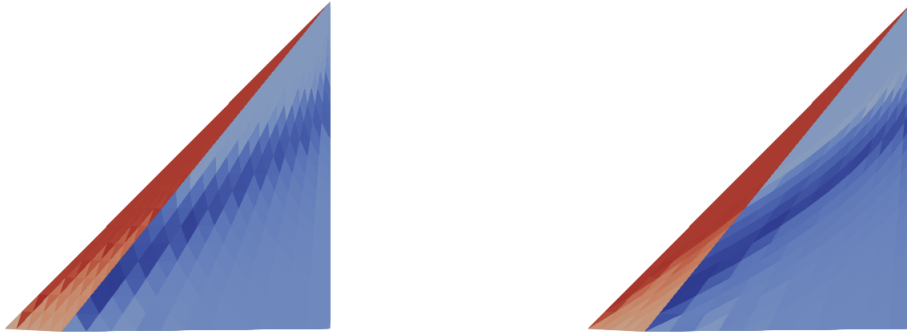


Fig. 9.39: Top view of the coarsest delta wing mesh used in MachLine to compare against PAN AIR. For use in MachLine, each quadrilateral panel has been split into two triangles. In both MachLine and PAN AIR, the mesh was mirrored across the root plane.

For both MachLine and PAN AIR, only a half mesh was generated, and mirroring across the xz plane was specified in the input to both.

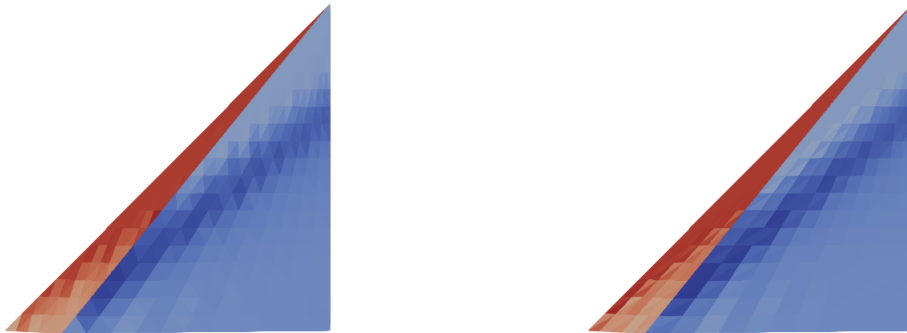
First the effect of dividing each quadrilateral panel into two triangles was examined. The pressure distributions over the top of the wing at 0° angle of attack using both methods of splitting the panels are shown in Figs. 9.40 and 9.41. From these, it is clear that the pressure distributions were better for both lower- and higher-order distributions when the panels were split along the root-leading-edge-to-tip-trailing-edge diagonal. This was likely due to the fact that these diagonals were more closely aligned with the Mach cones, leading to a truer representation of the flow phenomena. Based on this study, the MachLine meshes were generated with the panels subdivided along this diagonal.

Once the proper meshing scheme was determined, the grid convergence of both PAN AIR and MachLine was estimated following the procedure outlined in Section 8.2. This was done for $M_\infty = 1.62$ and angles of attack ranging from 0 to 5° . The results over angle of attack are shown in Fig. 9.42. The orders of convergence varied widely over all angles of attack for both PAN AIR and MachLine.



(a) Panels subdivided along root-trailing-edge-to-tip-leading-edge diagonal. (b) Panels subdivided along root-leading-edge-to-tip-trailing-edge diagonal.

Fig. 9.40: Pressure distributions over the delta wing using the medium mesh at 0° angle of attack using lower-order singularity distributions showing the differences due to which diagonal was used to subdivide each quadrilateral panel for MachLine. The color scale indicates high pressures with dark red and low pressures with dark blue.



(a) Panels subdivided along root-trailing-edge-to-tip-leading-edge diagonal. (b) Panels subdivided along root-leading-edge-to-tip-trailing-edge diagonal.

Fig. 9.41: Pressure distributions over the delta wing using the medium mesh at 0° angle of attack using higher-order singularity distributions showing the differences due to which diagonal was used to subdivide each quadrilateral panel for MachLine. The color scale indicates high pressures with dark red and low pressures with dark blue.

In general, PAN AIR had a higher order of convergence in C_x , while MachLine tended to have higher orders of convergence for C_z . For both force coefficients, the higher-order singularity distributions performed particularly poorly. This was due to the distributions being sensitive to the paneling at the wingtips. At the wingtips on the finest mesh, the panels were very fine, which as noted before causes poor behavior with the higher-order

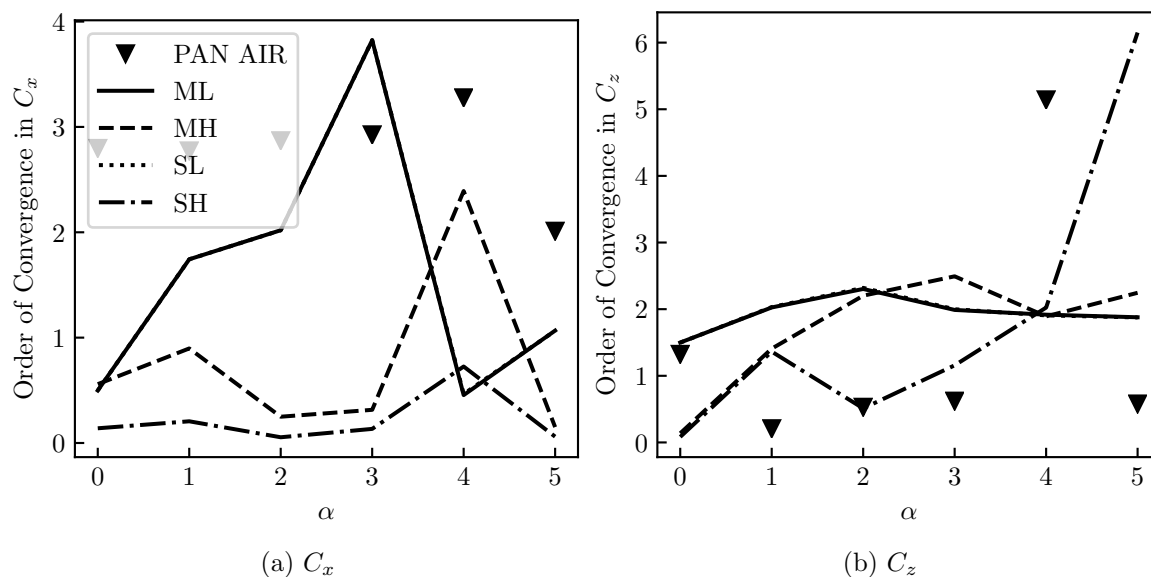


Fig. 9.42: Orders of grid convergence using PAN AIR and the various formulations in MachLine for the delta wing as a function of angle of attack.

singularity distributions. As shown in Fig. 9.43, the predicted pressures near the wingtip were in error when the higher-order distributions were used. This was likely the cause for the poor grid convergence of the higher-order singularity distributions. Since PAN AIR uses higher-order singularity distributions without having such issues appear, this highlights the potential for significantly improving the higher-order singularity distributions used in MachLine.

Next, the total force coefficients (axial and normal) were compared between MachLine and PAN AIR. For both MachLine and PAN AIR, the force coefficients were calculated using the second-order pressure rule, which is the default in PAN AIR. These force coefficients are shown using the finest mesh in Fig. 9.44. The force coefficients predicted by MachLine using the lower-order singularity distributions were very close to those predicted by PAN AIR at all angles of attack. When the higher-order singularity distributions were used, the predicted values from MachLine for C_z were reasonably close to those from PAN AIR. However, the results obtained for C_x using higher-order singularity distributions were poor. This was likely due to the paneling issue at the wing tips already discussed.

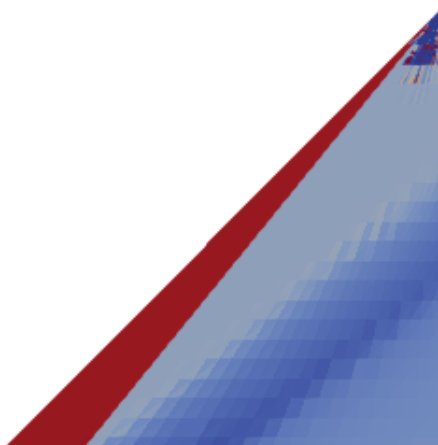


Fig. 9.43: Pressure results over the outer portion of the delta wing at 0° angle of attack using the higher-order singularity distributions and the Morino formulation showing the erroneous pressure predictions at the wingtip. The color scale ranges from $C_P \approx 1.84$ (dark red) to $C_P \approx -0.54$ (dark blue).

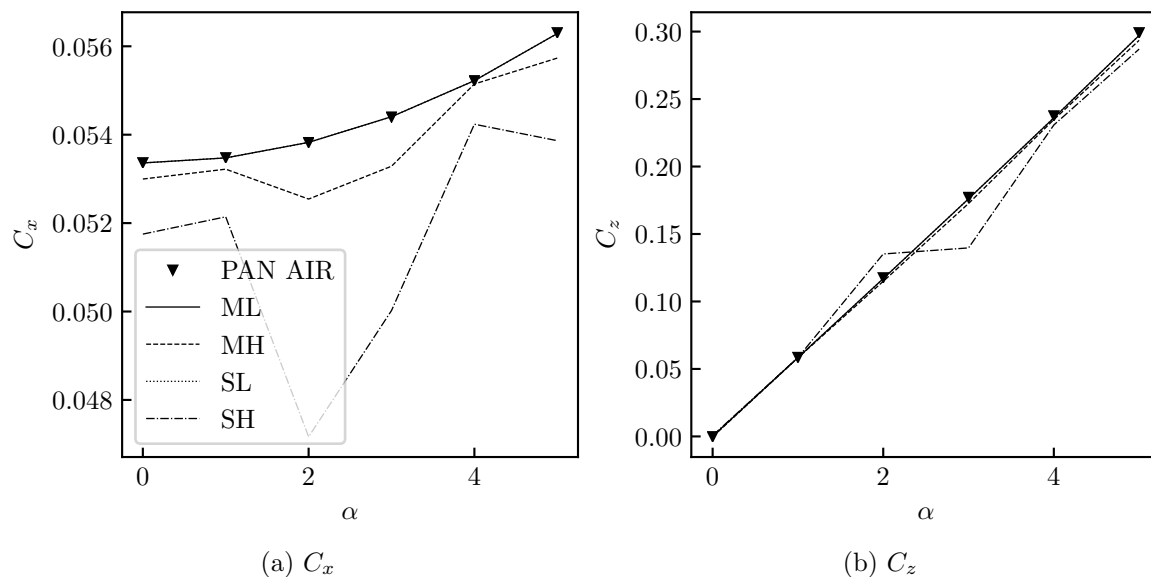


Fig. 9.44: Total (a) axial and (b) normal force coefficients acting on the delta wing (fine mesh) as a function of angle of attack predicted by PAN AIR and MachLine.

For comparison, the total force coefficients obtained from the medium mesh are shown in Fig. 9.45. For this mesh, the higher-order singularity distributions gave very good results, though the lower-order singularity distributions still produced results closest to PAN AIR. Over all angles of attack, the higher-order distributions resulted in lower values for C_x and

C_z than did either the higher-order distributions or PAN AIR. It may be noted that the differences between C_z predicted by MachLine using lower-order distributions and by PAN AIR were less with the fine mesh than with the medium mesh, indicating that both methods were converging to a similar result.

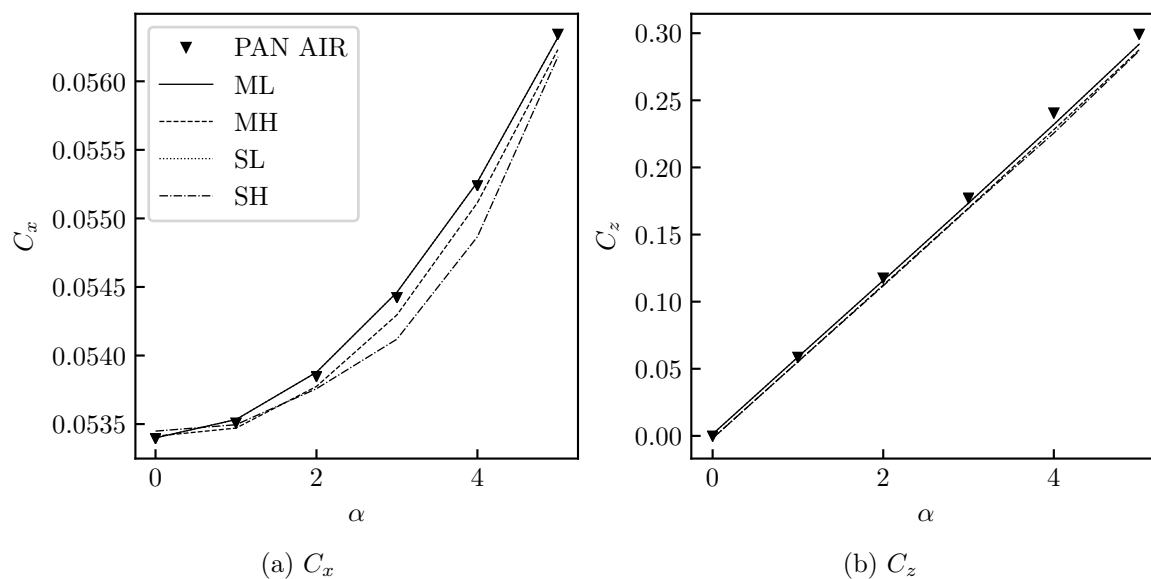


Fig. 9.45: Total (a) axial and (b) normal force coefficients acting on the delta wing (medium mesh) as a function of angle of attack predicted by PAN AIR and MachLine.

The predicted pressure distributions were then compared at the spanwise stations reported by PAN AIR. For each panel, PAN AIR reports the pressure coefficient at the panel centroid. Comparing this with the mesh used in MachLine is difficult, because the centroids of the PAN AIR panels fall on the midpoints of the dividing edges for the MachLine mesh. Thus, to allow for a more direct comparison, the results from MachLine were reported at these midpoint locations as the average the pressure coefficients on the two neighboring panels. The pressure distributions were compared using the isentropic pressure rule. Only selected results that were representative of all results obtained are presented here.

The results along a slice at 6% semispan at various angles of attack are shown in Fig. 9.46. At all angles of attack, the differences between PAN AIR and MachLine were very small. The most-significant exception to this was at $\alpha = 2^\circ$, where the pressures at the

trailing edge predicted using the higher-order singularity distributions were clearly in error. This was likely due to the trailing edge sensitivities discussed previously. Otherwise, the greatest discrepancies were seen just forward and aft of the line of maximum thickness. In these locations, the higher-order singularity distributions resulted in some oscillations in pressure coefficient, while the lower-order distributions gave results that matched PAN AIR very closely. These oscillations primarily seen with the higher-order singularity distributions were likely due to the heightened upstream influence introduced by the higher-order distributions. Despite this, all results from MachLine were remarkably accurate compared to PAN AIR.

The pressure results along a slice at 50% semispan at various angles of attack are shown in Fig. 9.47. These results were very similar to the ones seen at 6% semispan, including the erroneous pressures due to the higher-order singularity distributions at $\alpha = 2^\circ$. On the forward portion of the wing, all results from MachLine matched those from PAN AIR almost exactly. Over the rest of the wing, the greatest discrepancies appeared when the higher-order singularity distributions were used. These primarily appeared at sudden discontinuities in the pressure distribution, such as at about $x/c_r = 0.65$ in Fig. 9.47c. Just before the sudden drop in pressure due to the bow characteristic, the results from the higher-order distributions deviated from those predicted by PAN AIR and the lower-order distributions. Again, this is likely due to the heightened upstream influences introduced by the midpoint-interpolation method used for higher-order singularity distributions in MachLine. This type of behavior was not seen from PAN AIR, which also uses quadratic doublet panels, indicating this issue is specific to MachLine, not quadratic distributions in general.

The pressure results at 73% semispan are shown in Fig. 9.48. At this spanwise station, there were more differences between the results from MachLine and PAN AIR than have been seen previously. At all angles of attack, the pressure results from MachLine deviated from those from PAN AIR. The pressure results from MachLine deviated from PAN AIR most significantly at the trailing edge. This was true for the lower-order distributions, not

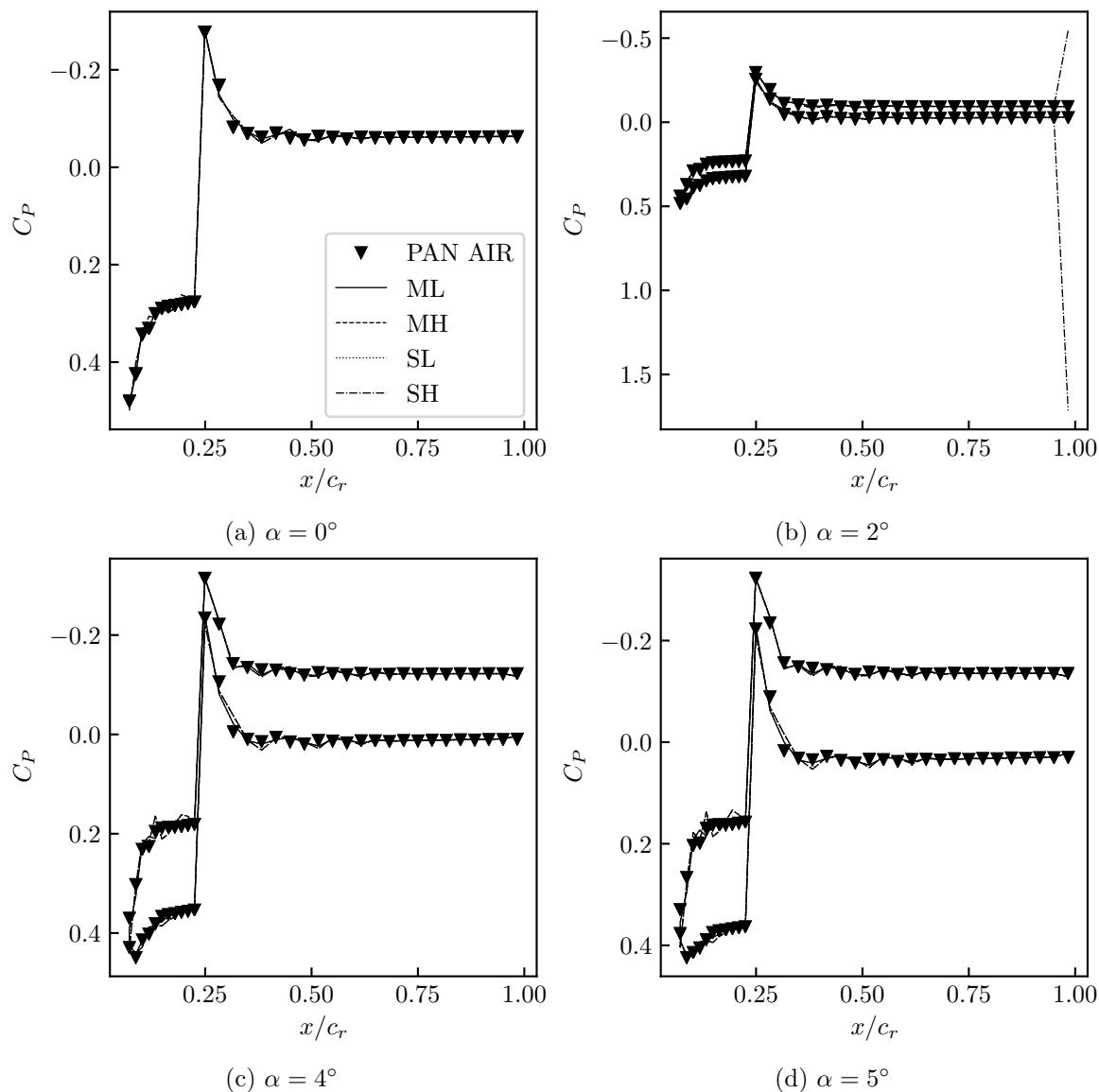


Fig. 9.46: Surface isentropic pressure coefficients at 6% semispan on the delta wing at varying angles of attack from MachLine and PAN AIR.

just the higher-order distributions as seen before. As before, the results from MachLine matched the results from PAN AIR almost exactly forward of the bow characteristic.

Finally, the total time taken to run MachLine and PAN AIR on equivalent meshes may be considered. These run times, averaged over all angles of attack and shown as a function of mesh resolution, are shown in Fig. 9.49. MachLine ran faster than PAN AIR by a significant margin for every case. This was likely due to the significant parallelism

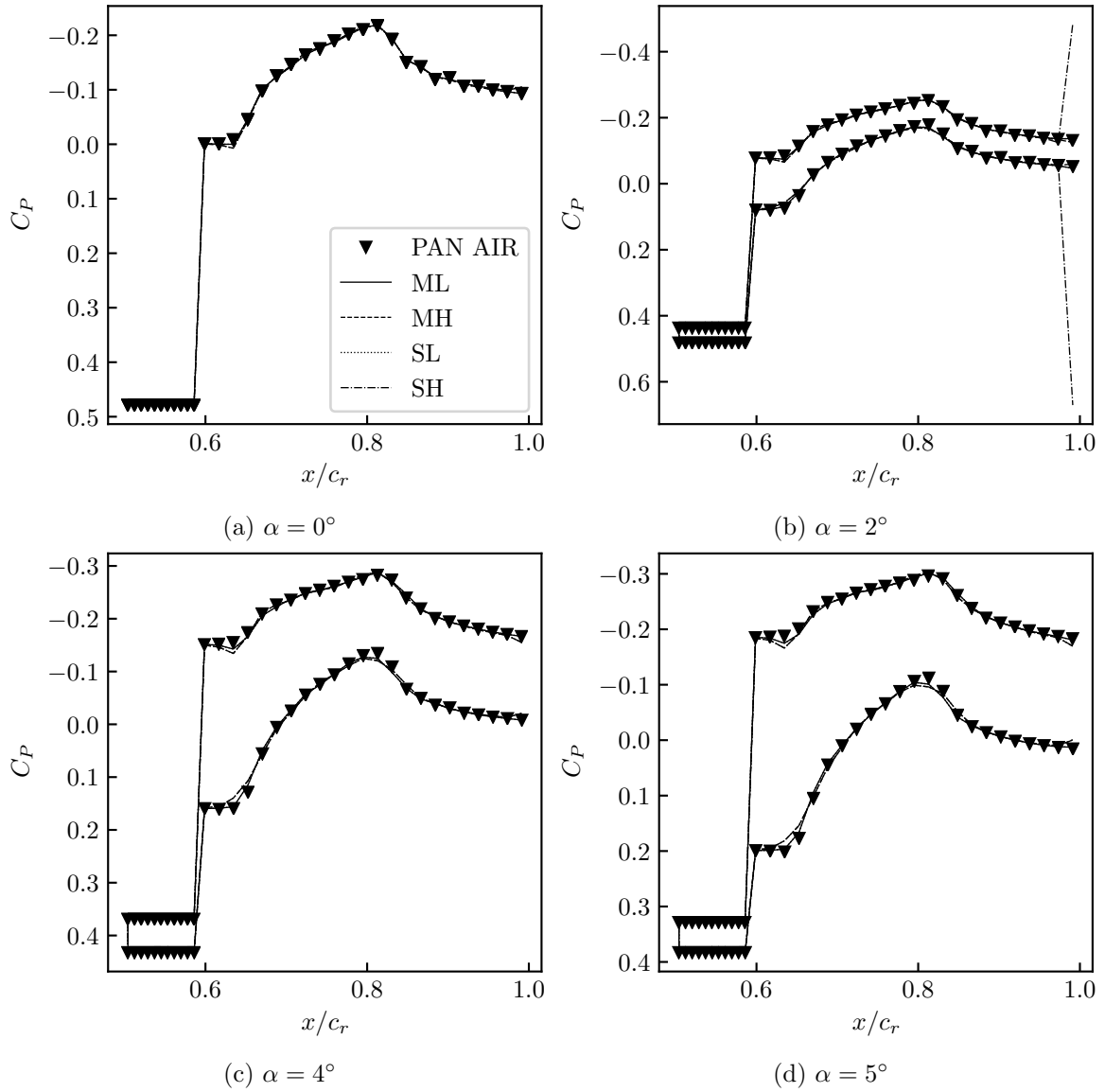


Fig. 9.47: Surface isentropic pressure coefficients at 50% semispan on the delta wing at varying angles of attack from MachLine and PAN AIR.

employed in the implementation of MachLine; PAN AIR has not been implemented at all in parallel. Between the MachLine results, the lower-order singularity distributions tended to run faster than the higher-order singularity distributions. There were no consistent differences in run time between the boundary condition formulations.

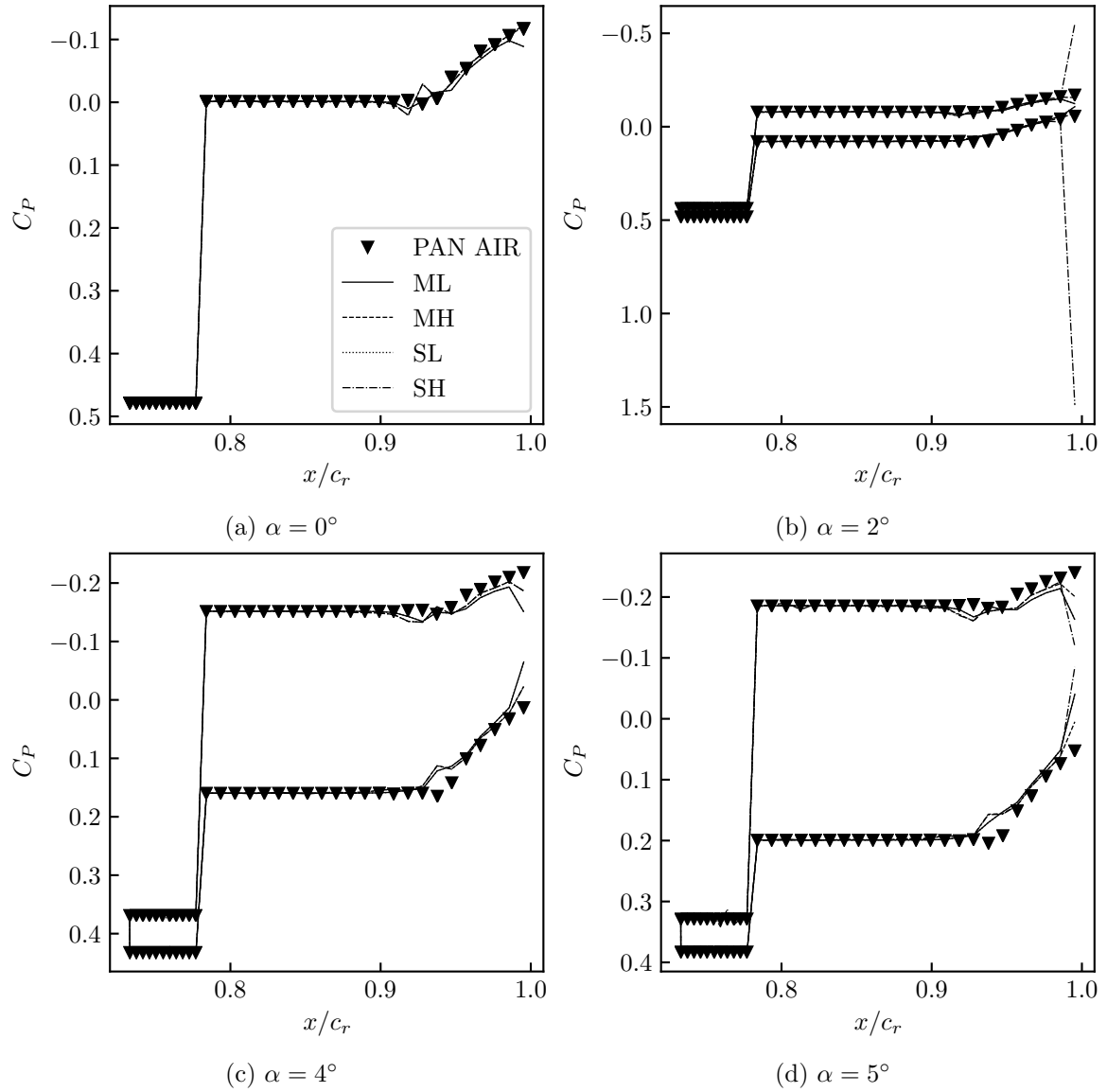


Fig. 9.48: Surface isentropic pressure coefficients at 73% semispan on the delta wing at varying angles of attack from MachLine and PAN AIR.

Overall, the results obtained from MachLine lined up very well with those obtained from PAN AIR. The greatest discrepancies were due to the weaknesses of the higher-order distributions already discussed.

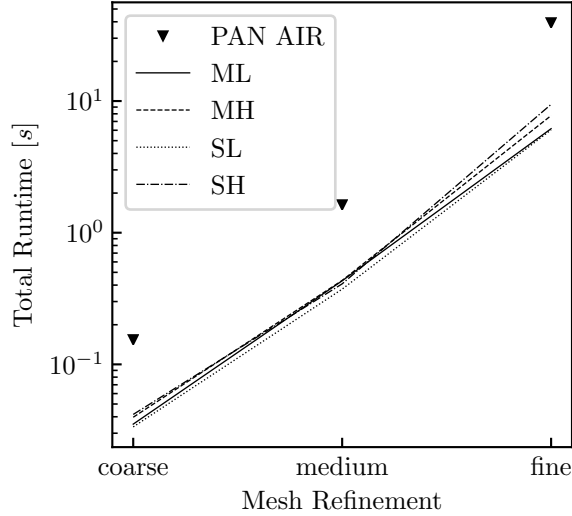


Fig. 9.49: Average run times for MachLine and PAN AIR as a function of mesh resolution for the supersonic delta wing.

9.6 Flow-Through Nacelles

Internal flows are known to be difficult to model using panel methods. This section presents a simple study into the mesh convergence of flow-through nacelles. Both subsonic and supersonic flow conditions are considered.

9.6.1 Subsonic Nacelle

An axisymmetric nacelle in incompressible flow was used to examine the differing effects of axial and circumferential mesh refinement on a flow-through nacelle. Since the flow was axisymmetric, it would be anticipated that the solutions were much less sensitive to circumferential refinement than axial refinement.

The nacelle used for this study had a NACA 0010 section with unit chord length. The diameter (chord line to chord line) of the nacelle was 0.5. Three different mesh densities were considered in both the axial and circumferential directions with the number of panels in the relevant direction doubled between each level of mesh refinement. What is meant by axial and circumferential refinement is shown in Figs. 9.50 and 9.51, respectively. While the mesh density was varied in one direction, it was held constant at the most refined level

in the other direction. The most-refined mesh in both directions had approximately 3200 panels.

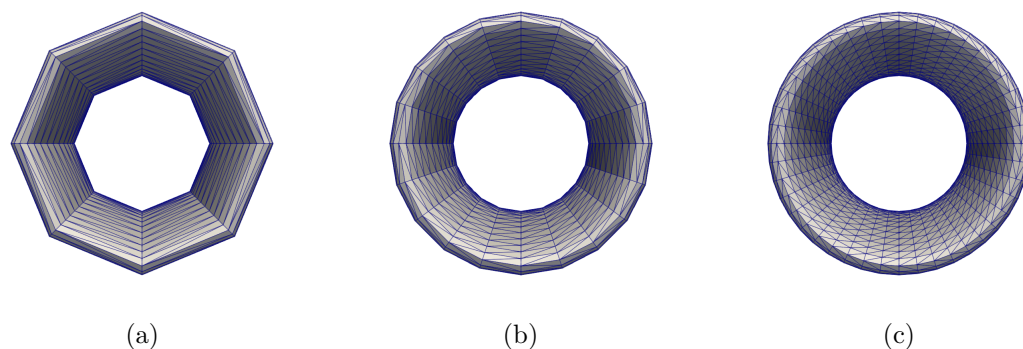


Fig. 9.50: Meshes showing effect of varying mesh refinement in the circumferential direction going from (a) coarse to (c) fine.

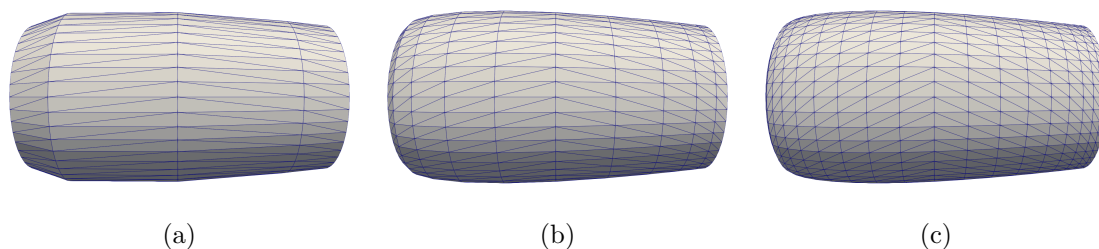


Fig. 9.51: Meshes showing effect of varying mesh refinement in the axial direction going from (a) coarse to (c) fine.

The surface pressure coefficient distributions at varying levels of axial mesh refinement are shown in Fig. 9.52. The results showed good convergence, though the results did vary significantly between the levels of mesh refinement, particularly on the inner surface of the nacelle where the pressure was lower. For varying axial refinement, the coarser meshes consistently over-predicted the pressure on the inner surface of the nacelle. On the outer surface of the nacelle, the pressure was rather insensitive to mesh density, only showing significant variation at the coarsest mesh level.

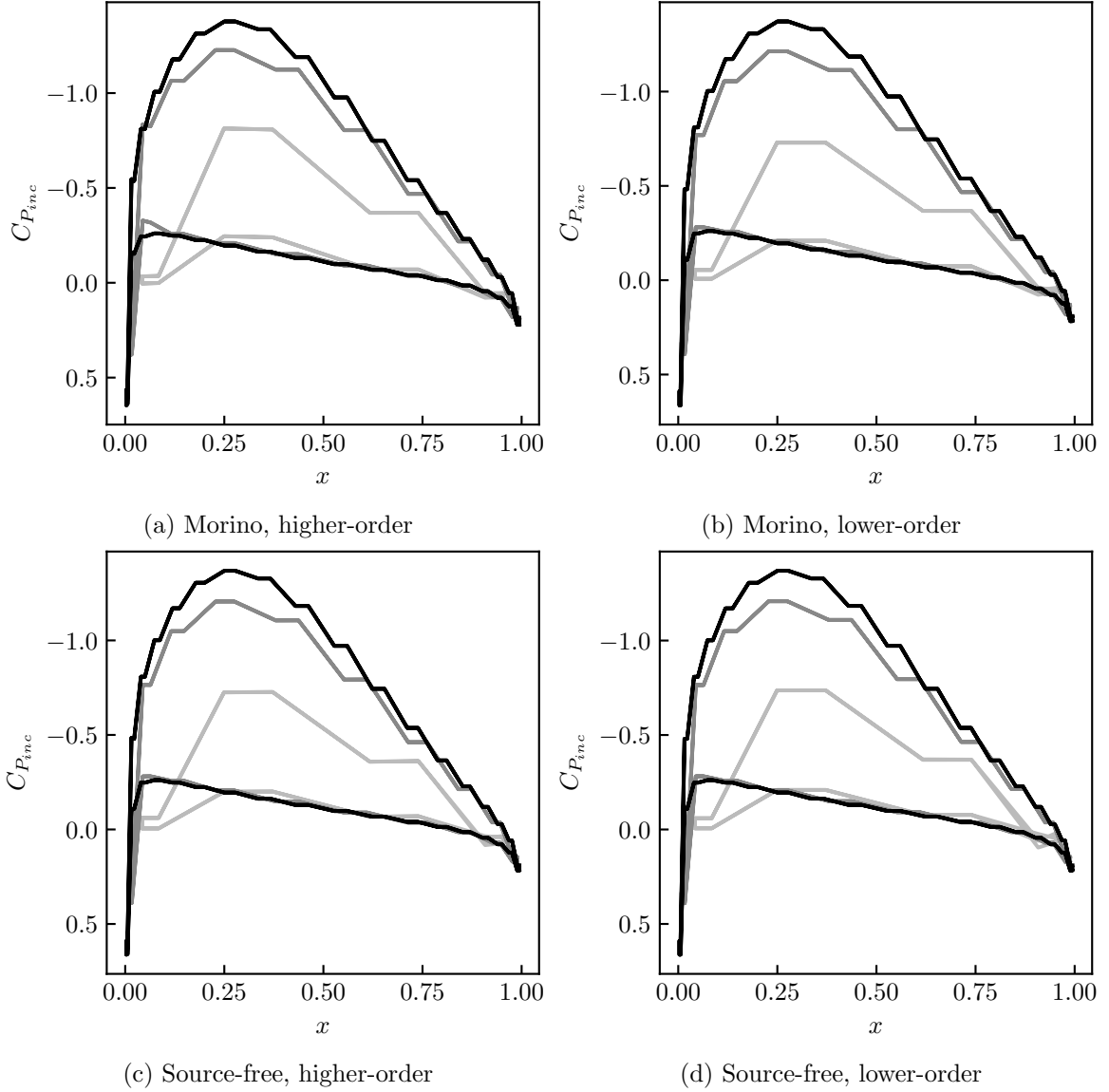


Fig. 9.52: Surface isentropic pressure coefficients along an axial slice of the NACA 0010 nacelle in incompressible flow for varying levels of axial mesh refinement. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).

In contrast, the surface pressure coefficients at varying levels of circumferential mesh refinement are shown in Fig. 9.53. Grid convergence in this case was also very good. Unlike when the axial mesh refinement was varied, the coarser meshes here consistently under-predicted the pressure coefficient on the inner surface of the nacelle. This makes sense, because a mesh which is coarser in the circumferential direction will restrict the flow

through the nacelle more than the true shape (compare Figs. 9.50a and 9.50c). More flow restriction in incompressible flow means greater flow velocities and thus lower pressure coefficients. Conversely, on the outer surface of the nacelle, the pressure distribution was practically insensitive to circumferential mesh refinement.

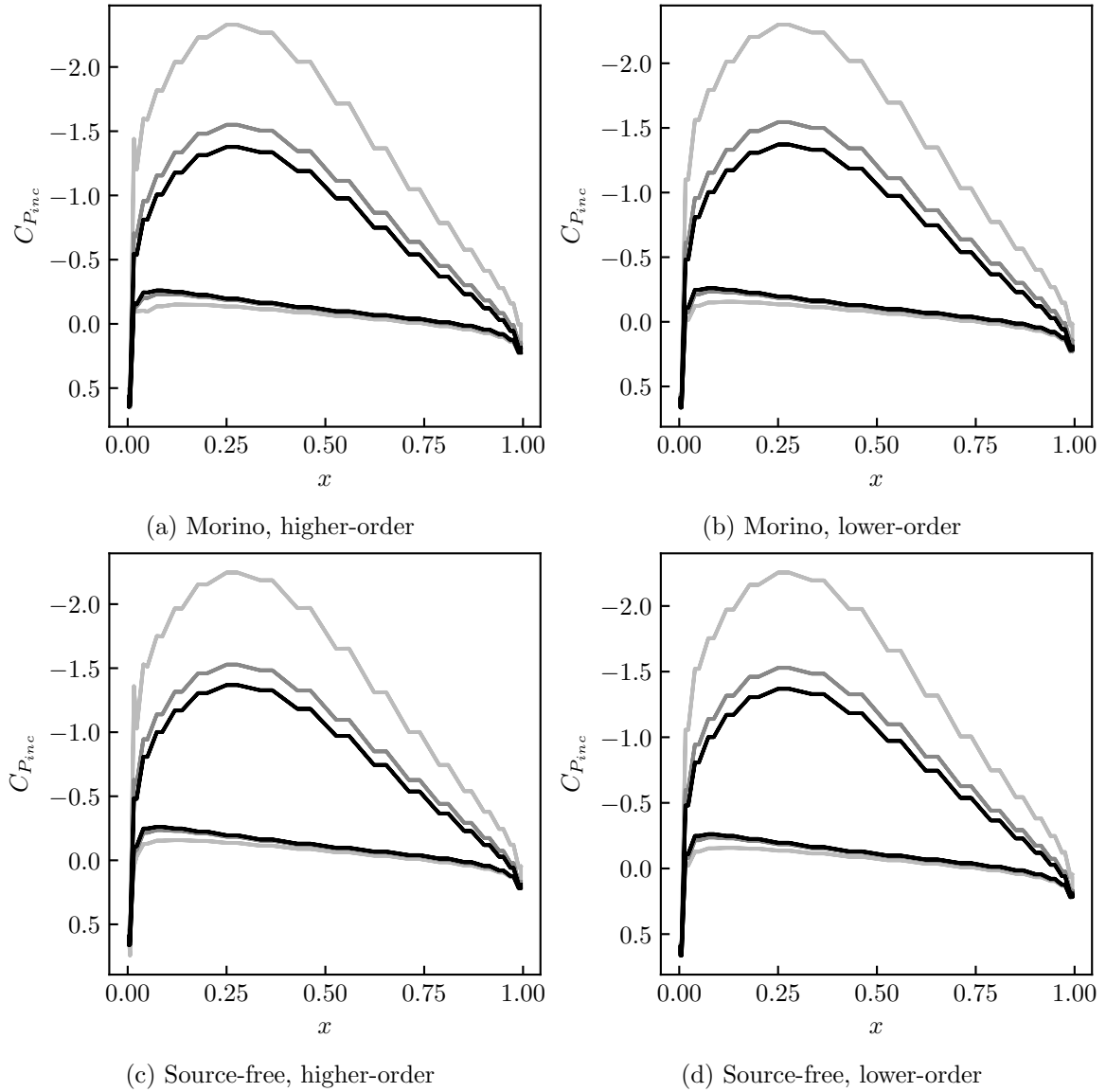


Fig. 9.53: Surface isentropic pressure coefficients along an axial slice of the NACA 0010 nacelle in incompressible flow for varying levels of circumferential mesh refinement. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).

Since a coarse mesh in the axial direction led to an over-prediction of the inner pressure coefficients, and a coarse mesh in the circumferential direction led to an under-prediction of the inner pressure coefficient, it was deemed necessary to examine the interplay of varying the mesh refinement in both directions simultaneously. The resulting pressure distributions are shown in Fig. 9.54. Here, grid convergence was even better than with the previous two cases, and the medium-density mesh resulted in pressure distributions nearly identical to the fine mesh. This highlights the need for balanced mesh refinement in most cases.

In all cases, the differences between the different boundary condition formulations and singularity distributions were small. Only when the mesh refinement was varied in both directions did significant differences appear. In this case, the Morino boundary condition formulation paired with higher-order distributions gave remarkably good results at the lowest refinement level.

9.6.2 Supersonic Nacelle

In supersonic flow, the problem of internal flow modeling is especially difficult due to the presence of Mach waves that reflect between the bounding surfaces of the internal region. The purpose of this study is to evaluate the ability of MachLine to produce grid-convergent results for cases where such Mach wave reflections are present and examine the effect of singularity distribution and boundary condition formulation on the solutions.

For this study, an axisymmetric (about the x -axis) nacelle with a triangular section was considered. The outer surface of the nacelle section was flat and aligned with the x -axis. The outer diameter of the nacelle was 0.5, and the nacelle had unit chord length. The maximum thickness of the nacelle section was 0.05, located at $x = 0.25$. Four different mesh resolutions were used, ranging from approximately 600 (coarse) to approximately 50000 panels (fine). The nacelle was tested at $M_\infty = 1.5, 2$, and 2.5 , which allowed for examining Mach wave reflections at various axial locations.

Since the outer surface of the nacelle was everywhere parallel to the flow, the surface pressure coefficient should be identically zero on the outside of the nacelle, regardless of

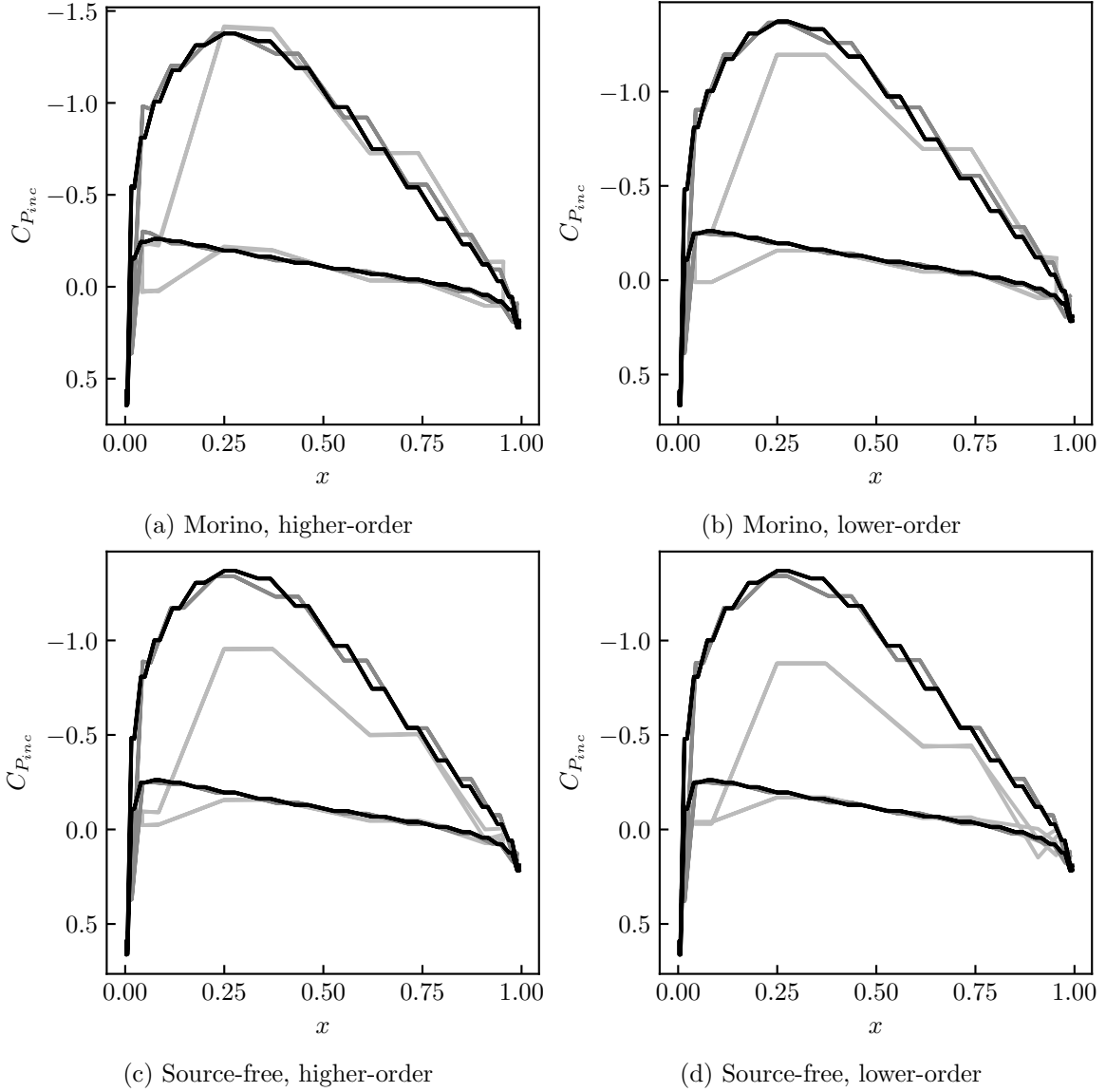


Fig. 9.54: Surface isentropic pressure coefficients along an axial slice of the NACA 0010 nacelle in incompressible flow for varying levels of mesh refinement in both the circumferential and axial directions. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).

M_∞ . The inner pressures should vary significantly with axial location, showing a sharp change at the location of maximum thickness and at the Mach wave impingement locations.

The inner surface pressures at $M_\infty = 1.5$ are shown in Fig. 9.55. For all boundary condition formulations and singularity distributions, the results showed good convergence with increasing mesh density. There was a sharp drop in pressure coefficient at the location

of maximum thickness, as expected. In addition, two Mach wave reflection locations may be clearly seen at $x \approx 0.5$ and $x \approx 0.75$. The lower-order distributions did result in some pressure spikes at the very trailing edge of the nacelle. However, these went away with increasing mesh density.

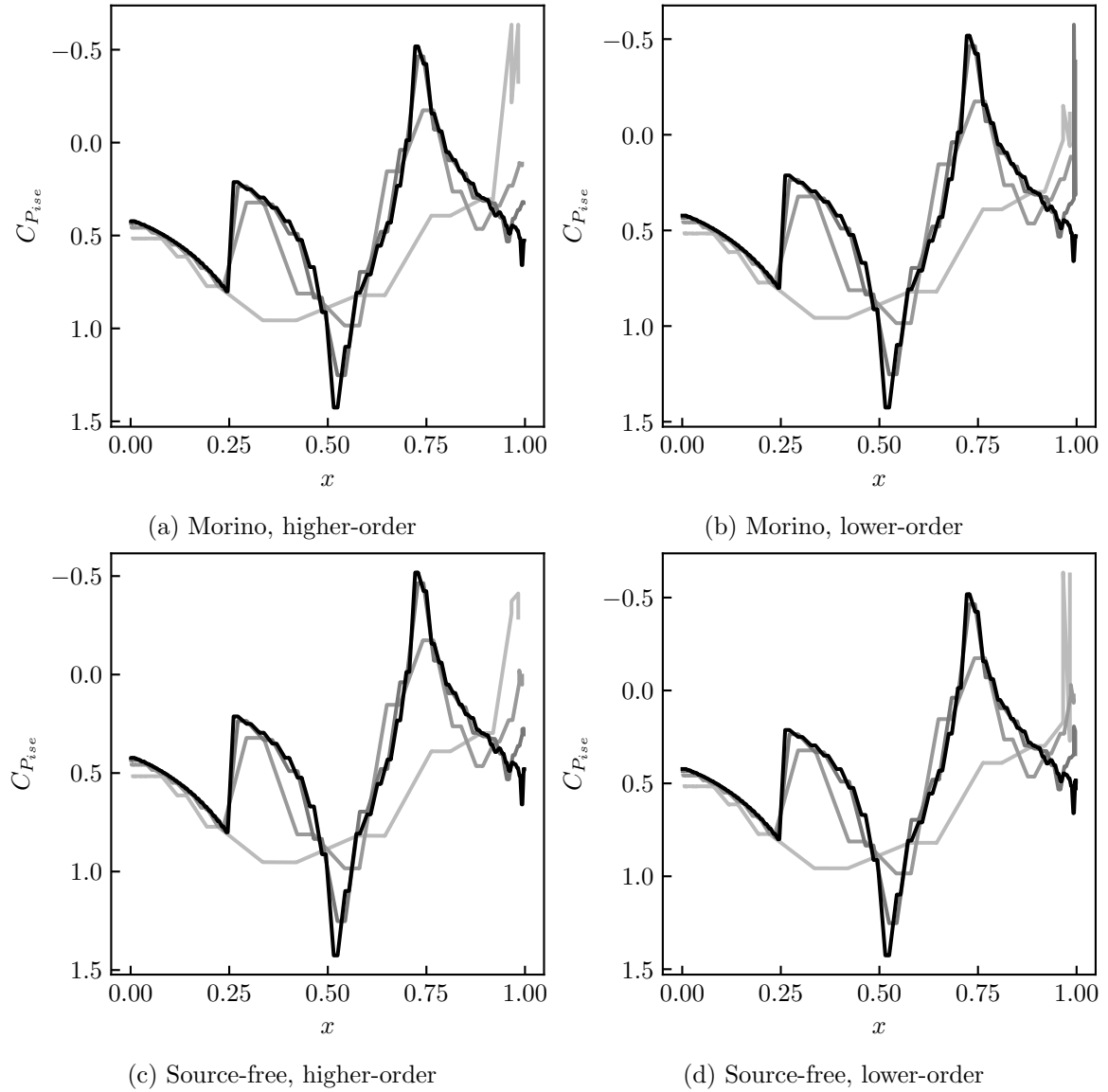


Fig. 9.55: Surface isentropic pressure coefficients along an axial slice of the inside of the supersonic nacelle at $M_\infty = 1.5$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).

The outside pressures at $M_\infty = 1.5$ are shown in Fig. 9.56. As mentioned previously, the surface pressure coefficient should be everywhere zero. For all boundary condition formulations and singularity distributions, the pressure coefficient everywhere converged toward zero as the mesh was refined. The most variation in pressure coefficient was seen just aft of $x \approx 0.5$. As shown in Fig. 9.55, a Mach wave hit the inside surface at about $x \approx 0.5$. The pressure fluctuations seen on the outside of the nacelle were likely due to this Mach wave not being entirely cancelled out by the inside surface sources and doublets. This is simply a product of the boundary integral equation being solved using discrete panels with approximate source and doublet distributions. Even so, the outer pressure variations were minimal on the finest mesh.

The inside surface pressures at $M_\infty = 2$ are shown in Fig. 9.57. As with the $M_\infty = 1.5$ case, both grid convergence and qualitative behavior were very good. Due to the higher Mach number (which led to a shallower Mach angle), there was only one location inside the nacelle where a Mach wave hit the inside surface. Results between the different boundary condition formulations and singularity distributions were again nearly identical, with the lower-order distributions resulting in some small spikes in pressure near the trailing edge when the coarser meshes were used.

The outside surface pressures at $M_\infty = 2$ are shown in Fig. 9.58. Again, grid convergence was good. Over most of the nacelle, the predicted pressure coefficient was very close to zero. Near the trailing edge, however, there was a region of oscillations in pressure coefficient. Comparing this with Fig. 9.57 reveals that (as with the $M_\infty = 1.5$ case) these disturbances were just aft of where a Mach wave hit the inside surface. The oscillations quickly died out, rather than increasing in amplitude as one moves downstream, which is very good. However, similar oscillations appeared in the inner pressure distributions (Fig. 9.57) at about the same location. Since the source and doublet distributions on both the inside and outside surfaces influence each other, it is likely that the oscillations on the outside surface induced similar oscillations on the inside surface.

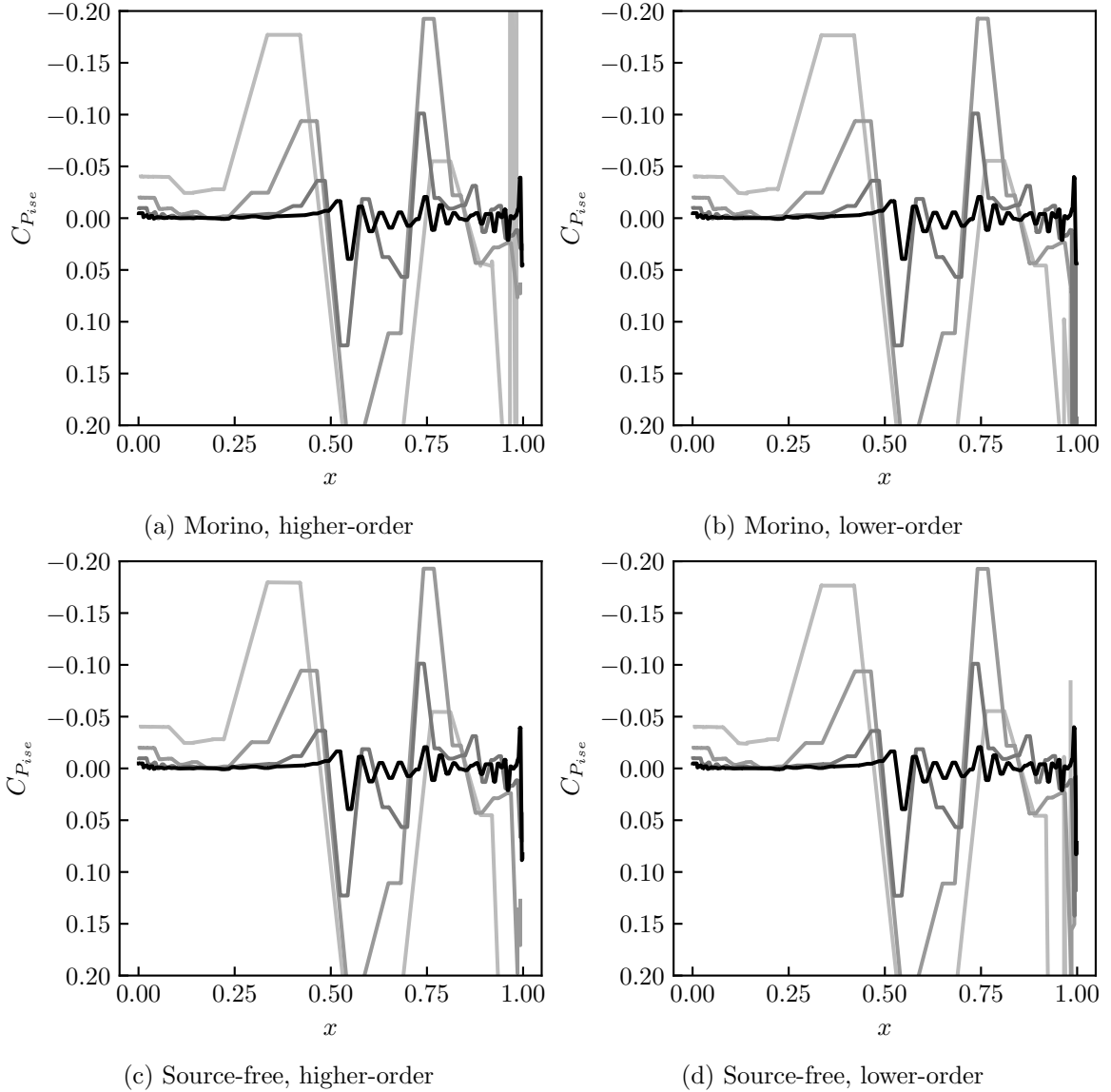


Fig. 9.56: Surface isentropic pressure coefficients along an axial slice of the outside of the supersonic nacelle at $M_{\infty} = 1.5$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).

To examine possible causes for these oscillations, the surface pressures at two slightly different Mach numbers ($M_{\infty} = 1.9$ and 2.1) were evaluated. The oscillations also appeared at $M_{\infty} = 1.9$, but were slightly smaller in magnitude and died off more quickly. At $M_{\infty} = 2.1$, the oscillations were not present at all. It is likely that the oscillations appeared due to a coincidental interaction of the mesh geometry with the Mach waves. However, these

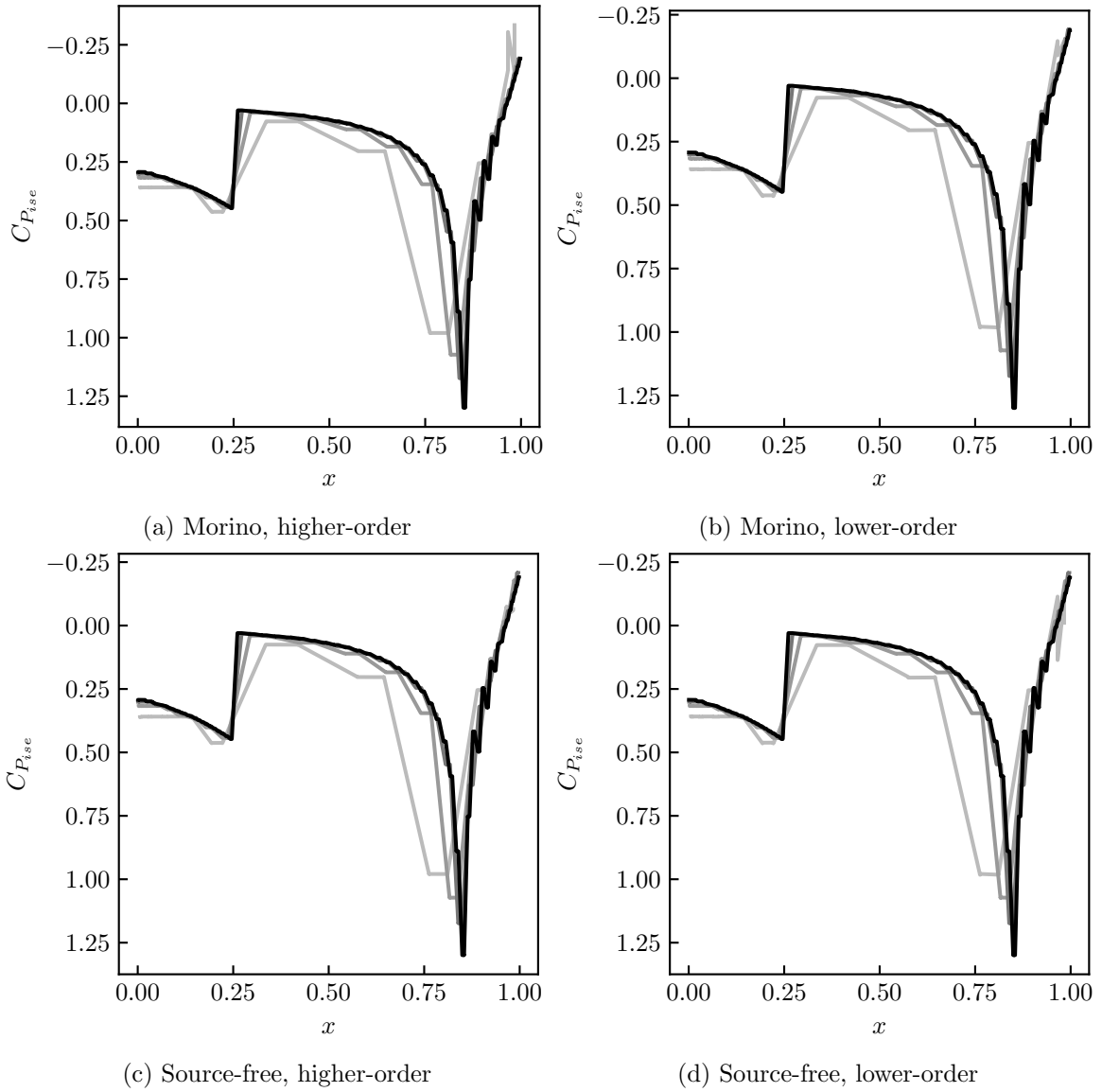


Fig. 9.57: Surface isentropic pressure coefficients along an axial slice of the inside of the supersonic nacelle at $M_{\infty} = 2$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).

oscillations died off quickly and were relatively small in magnitude, and so were not a cause for concern.

The inside surface pressures at $M_{\infty} = 2.5$ are shown in Fig. 9.59. In this case, there were no Mach waves impinging on the inside of the nacelle, and so the results were smooth,

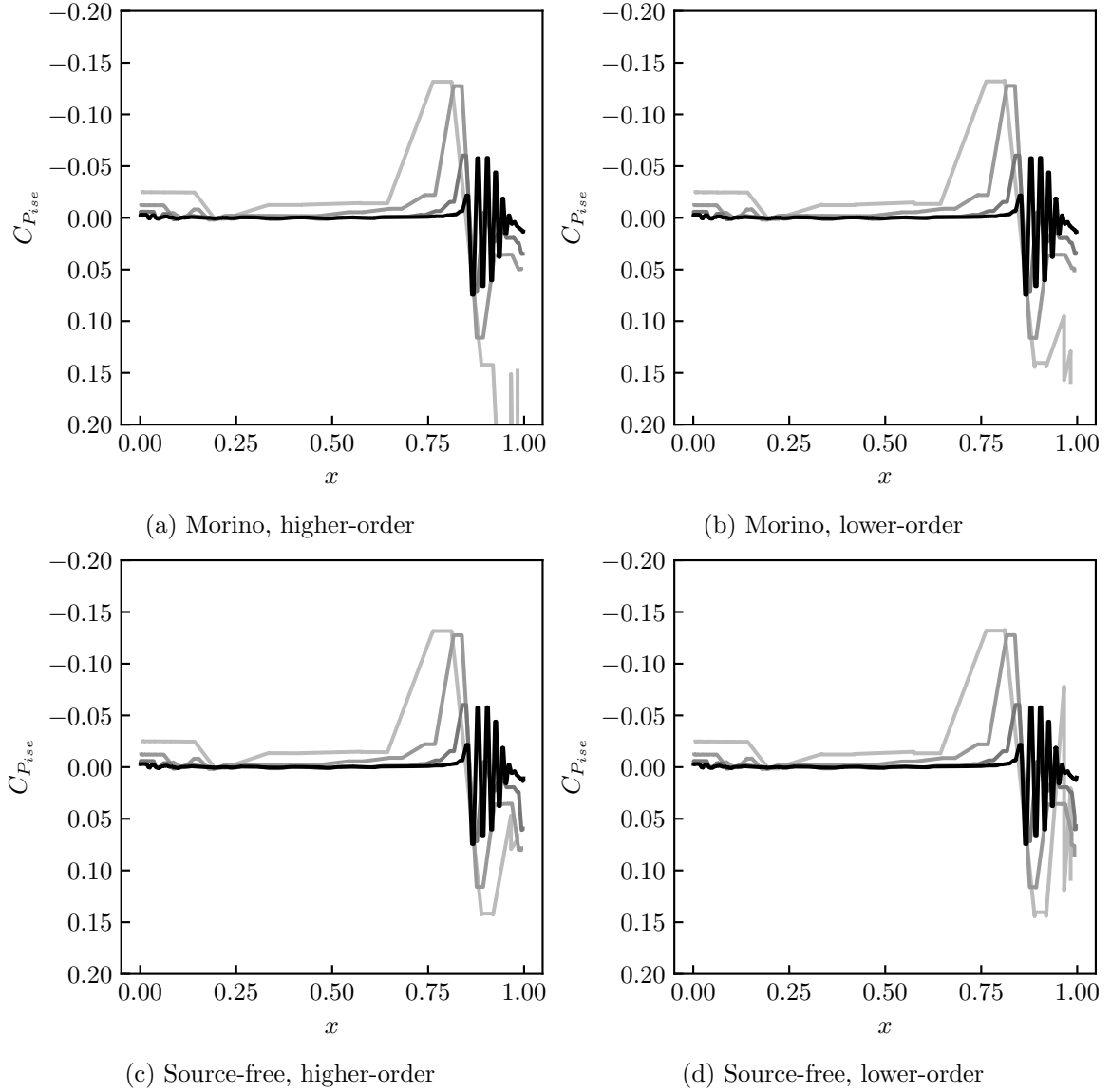


Fig. 9.58: Surface isentropic pressure coefficients along an axial slice of the outside of the supersonic nacelle at $M_\infty = 2$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).

aside from the sharp decrease in pressure at the location of maximum thickness. Grid convergence in this case was still very good.

The outside surface pressures at $M_\infty = 2.5$ are shown in Fig. 9.60. The results were everywhere very close to zero, likely due to a lack of internal Mach wave reflections, and grid convergence was good.

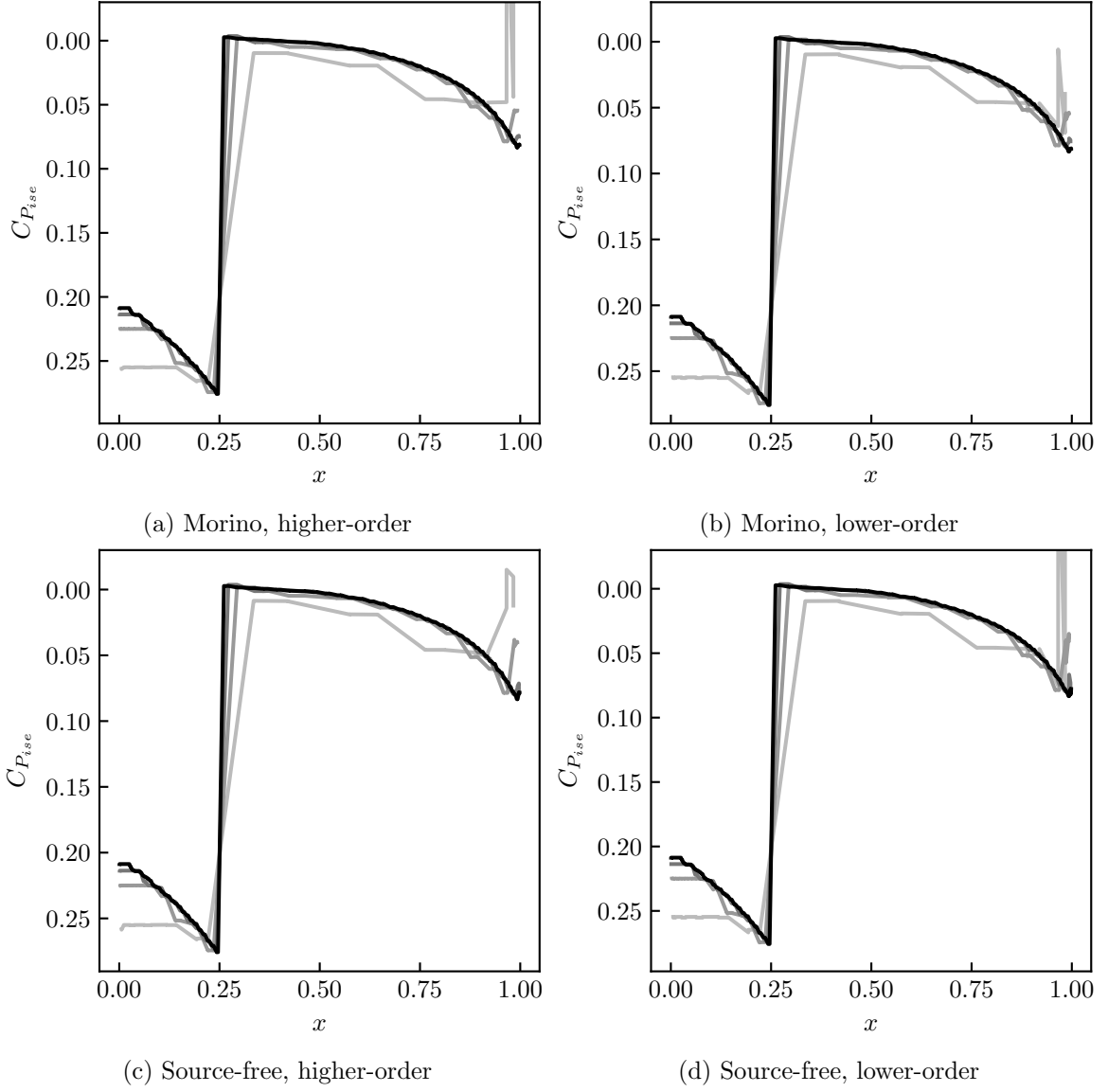


Fig. 9.59: Surface isentropic pressure coefficients along an axial slice of the inside of the supersonic nacelle at $M_{\infty} = 2.5$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).

9.7 Discussion

Overall, the pilot code has been shown here to be accurate for a wide variety of flows. Having presented all the evaluation cases, the overall performance of the singularity distributions and boundary condition formulations may be assessed.

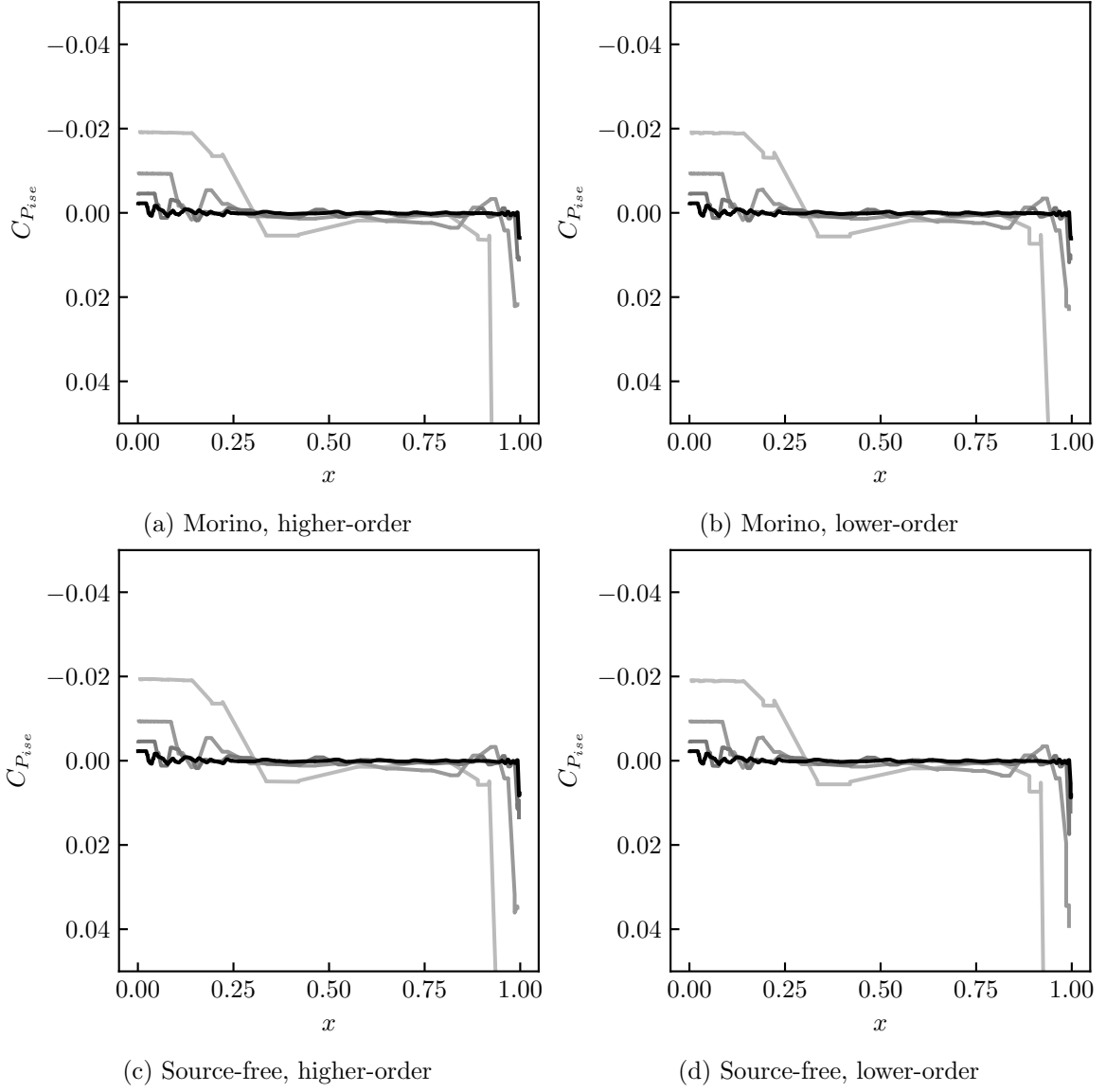


Fig. 9.60: Surface isentropic pressure coefficients along an axial slice of the outside of the supersonic nacelle at $M_\infty = 2.5$. The line shading represents the relative mesh densities, varying from coarse (light gray) to fine (black).

9.7.1 Singularity Distributions

The lower-order singularity distributions were most reliable at producing stable and accurate results. In no case were the results obtained using the lower-order singularity distributions poor nor did they demonstrate unique numerical instability.

In almost all cases, the results obtained using the lower-order singularity distributions were not affected by the choice of boundary condition formulation. This made sense with the nature of the source distributions. As discussed previously, the appropriate boundary condition for zero linearized mass flux through a panel (Eq. (2.10)) calls for the conormal derivative of the perturbation velocity potential to be constant on each panel if the panel is flat. This is exactly satisfied for the Morino formulation with constant-source panels because the inner perturbation potential is constant and the source strength (i.e. the jump in conormal derivative) is also constant. The conormal derivative of perturbation velocity potential will also be constant on each panel for the source-free formulation because the internal perturbation potential is linear. Because of this, the derivative of the inner perturbation potential in any direction is constant, and there are no sources to alter this at the surface. Thus, both the Morino and source-free formulations produced identical results when the lower-order singularity distributions were used, except in the subsonic nacelle convergence study (compare Figs. 9.54b and 9.54d). The reason for this is not known.

On the other hand, the higher-order distributions were less reliable than the lower-order distributions. In particular, the higher-order distributions were shown to be particularly sensitive to sudden qualitative shifts in the mesh and extremely low aspect ratio panels. In addition, the forcing of linearity at trailing edges often caused erroneous results.

However, as was seen in Section 9.5, the higher-order distributions resulted in smoother and more accurate pressure distributions for some configurations in supersonic flow. Thus, there are some advantages to the higher-order distributions.

Concerning the speed of MachLine, across all cases considered here, the lower-order singularity distributions generally resulted in faster run times than the higher-order singularity distributions. This made sense with the implementation of the different distributions presented in Chapter 5, as all calculations performed for the lower-order singularity distributions must also be performed for the higher-order singularity distributions, and then some. However, if the higher-order distributions could be improved to provide greater accuracy, then this increased computation time could offset by using a less-refined mesh.

9.7.2 Boundary Condition Formulations

As discussed previously, there were rarely differences between the two boundary condition formulations when the lower-order singularity distributions were used. When paired with higher-order singularity distributions, the Morino formulation tended to produce better results than the source-free formulation. In general, the Morino formulation produced results with less mesh sensitivity and greater accuracy.

On the other hand, the source-free boundary condition formulation tended to perform poorly when paired with higher-order singularity distributions. The mesh sensitivity of the results was greater and accuracy suffered. This is consistent with what has been proposed in the literature, that boundary condition formulations without sources are more sensitive to variations in the mesh than those with sources [50].

In terms of speed, the source-free formulation tended to slightly outperform the Morino formulation in terms of speed. This was consistent when lower-order singularity distributions were used. However, when higher-order distributions were used, the relative speed performance of the two boundary condition formulations was mixed. Overall, the choice of singularity distribution had a much greater impact on solution speed than did the choice of boundary condition formulation.

CHAPTER 10

CONCLUSION

This work has presented the theory and implementation of a modern, subsonic/supersonic panel method. The panel method as described here constitutes a viable alternative to Euler and Navier-Stokes solvers within the scope of preliminary aircraft design and optimization, particularly for supersonic aircraft. Panel methods have previously been overlooked for such tasks due to the limitations of existing legacy codes developed decades ago. In this work, it has been shown how a panel method may be implemented using modern computational techniques such as unstructured meshes and fast matrix solvers. In addition, the traditional assumptions regarding the need for sources and required order of singularity distributions have been reexamined. Much valuable information from the literature has been summarized and clarified as well, paving the way for the further development of modern compressible panel methods.

10.1 Summary of Work

First, the theory of compressible flows and panel methods, including their history, was reviewed in Chapter 1. In addition, the state of the art of supersonic panel methods was summarized, and weaknesses and areas for improvement were assessed. Based on these, four research objectives were identified (see Section 1.7).

It was then shown in Chapter 2 how internal Dirichlet boundary condition formulations may be used to determine the unknown surface source and doublet distributions. Of the two formulations considered here, one was the standard Morino formulation, and the other was a novel formulation that did not require surface source distributions for fully-subinclined configurations. This formulation, called the source-free formulation, was derived in Section 2.1.2. The advantage to not having sources was that source strengths and influences never need to be calculated, leading to some computational savings. Also, the implementation

of boundary conditions on superinclined panels (panels that fall entirely outside their own domain of dependence) in supersonic flows was explored in Section 2.2. The full implementation of superinclined panels was beyond the scope of this work. However, a theoretical discussion of how they could possibly be handled was presented.

Then, the processing of an unstructured mesh for use within an unstructured panel method was presented in Chapter 3. This involved describing the coordinate systems used, how adjacent panels are located and their edges characterized, and how meshes may be mirrored for computational efficiency. Most importantly, a robust procedure for placing control points inside an unstructured mesh was developed in Section 3.7. It was also discussed how simple wakes may be modeled automatically and efficiently.

Next, it was shown in Chapter 4 how domains of dependence may be efficiently calculated on an unstructured mesh. This required calculating how vertices, edges, and panels fall relative to any given domain of dependence. For determining the vertex domain of dependence relationships, two different methods were considered: a brute-force search and a novel recursive search. In comparing the two, the recursive search was shown in Section 4.3 to have reduced order of complexity compared to the brute-force search for slender configurations, even up to high Mach numbers. However, the additional memory requirements of the recursive method developed here, its inability to be parallelized, and the fact that it does not directly calculate the needed information made it less desirable for implementation in a supersonic panel method than the brute-force method.

Then, the use of singularity distributions of varying order was discussed in Chapter 5. It was shown in Section 5.2 how the perturbation velocity potential due to a panel with a given distribution of source and doublet strength may be calculated. This brought together and greatly clarified information from several literature sources. Novel methods for determining higher-order (quadratic doublets and linear sources) singularity distributions on an unstructured mesh were then developed and assessed in Sections 5.4 and 5.5. These methods involved interpolating the doublet and source strengths from known points on the mesh to determine the distribution of these strengths on a given panel.

The efficient solution of the linear system of equations was then considered in Chapter 6. It was shown how LU decomposition, the most common method for solving the linear system of equations in a supersonic panel method, performs very poorly in terms of speed when compared with other available solvers. In particular, GMRES was shown to be particularly fast and robust for solving the linear system of equations in serial compared to the other solvers considered. In addition, a set of novel matrix solvers were developed in Section 6.2.4 that exploit the large number of zero elements in a supersonic AIC matrix. These solvers (the QRUP and FQRUP solvers) performed comparably with GMRES when the system matrix was sorted into upper-pentagonal form. A novel, geometry-based algorithm for sorting the system into upper-pentagonal form was also developed in Section 6.1.

Then, the necessary formulas for post-processing the solution from a linear panel method were given in Chapter 7. These included formulas for calculating surface velocities and pressures, and the resultant forces and moments acting on the entire mesh. As with the influence calculations presented in Chapter 5, this involved bringing together and clarifying information available in the literature.

The numeric behavior of the panel method as implemented were then examined in Chapter 8. The specific attributes considered were sensitivity to control point offset (Section 8.1), order of grid convergence (Section 8.2), and sensitivity to panel regularity (Section 8.3). For all of these, the effect of boundary condition formulation and singularity distribution order were considered. Most importantly, the control point offset, k_1 , which is a free parameter, was found to have a large range (10^{-10} to 10^{-4}) over which it had negligible impact on solutions for all singularity distributions and boundary condition formulations. In terms of grid convergence and sensitivity to panel regularity, the lower-order singularity distributions were found to be the most robust with few differences appearing between the boundary condition formulations. Only with the higher-order singularity distributions were the boundary condition formulations found to give differing results. The higher-order distributions also tended to result in greater mesh sensitivity and a lower order of grid convergence. However, when higher-order distributions were used for configurations with

perfectly flat surfaces on which the flow properties varied, such as a double-wedge wing or a delta wing with a diamond airfoil section, the use of higher-order distributions showed a higher order of grid convergence for some cases.

Then in Chapter 9, the panel method was evaluated against available analytic and experimental data. The panel method was found to be remarkably accurate for small-perturbation flows outside of the transonic regime, as would be expected. For some configurations, the results predicted when using higher-order singularity distributions were smoother than those obtained when using lower-order singularity distributions (e.g. see Section 9.5). However, the higher-order singularity distributions again showed greater sensitivity to paneling (e.g. see Sections 9.2 and 9.4) and required more computation time. The lower-order singularity distributions were consistently accurate, robust, and fast. The differences between the boundary condition formulations were minimal. In general, when higher-order singularity distributions were used, the Morino formulation produced better results than the source-free formulation. The source-free formulation tended to be faster than the Morino formulation, but not by a significant margin.

10.2 Outcome of Objectives

In Chapter 1, the following four objectives were identified:

1. Characterize the impact of varying singularity distributions on the accuracy, speed, and robustness of an unstructured panel method.
2. Characterize the impact of boundary condition formulation on the accuracy, speed, and robustness of an unstructured panel method.
3. Assess the performance of various matrix solvers for the unique matrix equation appearing in supersonic panel methods.
4. Identify a viable path forward for the development of a recursive domain of dependence search algorithm and assess its impact on solution speed.

In order to meet the first objective, methods were developed in Chapter 5 for using both lower- and higher-order distributions on an unstructured mesh. The lower-order method is based off of that of Davis [7], with some improvements to allow for wakes. On the other hand, the method developed here for higher-order distributions is entirely novel. Overall as shown in Chapters 8 and 9, the lower-order distributions provided the best accuracy and were most robust (in terms of grid convergence and mesh sensitivity). However, the higher-order distributions did provide greater accuracy for some configurations. It appears that the effective use of the higher-order distributions requires carefully controlling the heightened continuity introduced between panels. In terms of speed, the higher-order distributions resulted in significantly longer run times than did the lower-order distributions. It must be noted that the relative behavior of the lower- and higher-order distributions was likely in large part specific to the implementation presented here. As shown when comparing to PAN AIR, which uses higher-order distributions, very good results can be obtained from higher-order distributions when carefully formulated.

In meeting the second objective, two boundary condition formulations (as derived in Chapter 2) were implemented in the pilot code. The first was the standard Morino formulation. The second was a novel source-free formulation. The source-free boundary condition formulation is a Dirichlet formulation, like the Morino formulation, but does not use sources. As demonstrated in Chapter 9, this made the source-free formulation faster in general than the Morino formulation, as source strengths and influences did not need to be calculated. Paired with lower-order singularity distributions, these two boundary condition formulations gave identical results. However, when higher-order singularity distributions were used, the results varied significantly between the two boundary condition formulations, with the Morino formulation typically giving better results. As before, this behavior could be different if the higher-order distributions were implemented differently.

To meet the third objective, six different matrix solvers (three direct and three iterative) were tested for solving the linear system of equations (Eq. (2.22)) and the results presented in Chapter 6. Two of the direct solvers considered (the QRUP and FQRUP solvers) were

novel algorithms derived in this work (see Section 6.2.4). These algorithms solved the system using the QR decomposition via Givens rotations. By way of a geometry-based sorting algorithm presented in Section 6.1, these novel solvers were made to exploit the many zero elements that appear in the supersonic AIC matrix. It was shown that LU decomposition, which has been standard for use with supersonic panel methods, was outperformed by every other solver considered. The iterative solvers were faster than the direct solvers for every case considered, with the GMRES solver being the fastest for most cases. However, on average, the BJAC and BSSOR solvers had the lowest order of complexity. In addition, the QRUP and FQRUP solvers outperformed all iterative solvers for some configurations. And, while originally developed for the novel direct solvers, the geometry-based sorting algorithm was shown to improve the performance of the BJAC and BSSOR solvers for one configuration.

To meet the fourth objective, a recursive domain of dependence algorithm was developed and tested. As shown in Chapter 4, compared to a brute-force approach to calculating domains of dependence, the recursive algorithm only provided a reduced order of computational complexity for extremely low aspect-ratio configurations at low Mach numbers. Otherwise, the recursive and brute-force methods showed similar performance. This, together with the fact that the recursive algorithm cannot be implemented in parallel, the memory requirements for the recursive algorithm, and the fact that it does not directly calculate the needed information made the recursive search algorithm inferior to a brute-force approach.

10.3 Recommendations for Users of MachLine

Significant effort has been expended to ensure the pilot code, MachLine, can be a usable tool beyond the scope of just this work. As such, it is appropriate to share with future users of MachLine lessons learned from this research effort.

Based on the research presented here, it is recommended to use the lower-order singularity distributions. These are more reliable than the higher-order singularity distributions, in general, and provide similar accuracy for less computational cost. If higher-order dis-

tributions must be used, the maximum continuity angle should be set carefully; ideally, it should be small.

If the lower-order distributions are used, then the choice of boundary condition formulation is not significant. The source-free formulation presents some advantages in terms of computational expense; otherwise, there is little difference between the two as currently implemented in MachLine. If higher-order distributions are used (again, not recommended), then the Morino boundary condition formulation should be used.

When choosing a matrix solver to use, several factors should be taken into account. The GMRES solver is very fast and accurate for the vast majority of cases, and so it is the default solver in MachLine. However, it should be remembered that the GMRES solver is not guaranteed to converge. If convergence becomes an issue, the FQRUP solver is a fast alternative that does not suffer from convergence issues, though it can produce somewhat higher residuals. For very large problems, the BJAC or BSSOR solvers may be valuable, as the block size and relaxation factor for these can be tuned to a specific case and computer to achieve very fast run times. As currently implemented, the BJAC solver within MachLine is significantly parallelized, and so should probably be chosen over the BSSOR solver unless experience indicates otherwise.

The default control point offset of 10^{-7} should work well for most cases. However, additional confidence in results may be gained by performing a control point offset study for a specific case.

To avoid spurious pressure results, the paneling near the tips of bodies of revolution should be as regular as possible. In general, paneling should be relatively even, and all panels should have low aspect ratio.

10.4 Novel Contributions

Throughout this work, several novel contributions to the field of compressible panel methods have been presented. These are summarized here in the order in which they appear in this work.

In Section 2.1.2, a new Dirichlet boundary condition formulation was derived. This formulation employs the traditional concept of zero normal linearized mass flux, but enforces it without using surface sources (unless superinclined panels are to be used). Being a Dirichlet formulation, this source-free formulation is based on enforcing certain values for the inner potential at a set of control points inside the body. To improve the usability of this formulation and the traditional Morino formulation, a new method for robustly placing control points inside an unstructured mesh was developed in Section 3.7. The accuracy and robustness of this source-free formulation was demonstrated in Chapters 8 and 9.

Some minor improvements to automatic wake modeling for unstructured meshes were also presented. In Section 3.5, a robust method for identifying wake-shedding edges was presented. A simple approach to efficiently truncating supersonic wakes was also given.

A novel algorithm for recursively determining domains of dependence was derived in Section 4.3. This algorithm was bench-marked against a brute-force algorithm and showed a lower order of time complexity for a low-aspect-ratio configuration at low Mach numbers.

A straightforward approach to implementing higher-order (quadratic-doublet-linear-source) singularity distributions on unstructured meshes was derived in Chapter 5. Multiple approaches were developed and tested. The final approach implements quadratic doublet distributions across each panel while not increasing the size of the problem to be solved and exactly enforcing continuity of doublet strength at all panel edges. While other approaches to implementing higher-order panels on unstructured meshes are possible, this is the first such attempt. The accuracy and robustness of these higher-order distributions were evaluated in Chapters 8 and 9.

In addition, Chapter 5 presented in a unified fashion all formulas necessary for calculating the perturbation potential induced by source and doublet panels at arbitrary points in

space for both subsonic and supersonic flows. Such a succinct and complete description of these calculations has been lacking in the literature regarding compressible panel methods.

In Chapter 6, the efficiency of various matrix solvers for solving the supersonic linear system of equations was considered for the first time in the literature regarding supersonic panel methods. A novel sorting algorithm was presented that, based only on the mesh geometry and freestream direction, enforces an upper-pentagonal structure to the AIC matrix. It was then shown how this novel structure may be exploited to solve the system of equations efficiently using QR decomposition.

10.5 Limitations

While it is hoped that the conclusions reached in this work may prove useful to others, it must be remembered that they were reached under certain assumptions and limitations. These will be summarized here.

As pointed out in Chapter 5, this work has been limited to the implementation of flat panels. This was done based on the assertion made in multiple literature sources that the influence coefficients for curved panels in supersonic flow cannot be evaluated [2, 50]. While this assertion may prove false under further study, it was deemed most efficient to place panel curvature outside the scope of this work and consider only the effects of singularity distribution and boundary condition formulation when flat panels were used. Should a method be determined for reliably evaluating the influence coefficients of curved panels in supersonic flow, some of the conclusions reached in this work may need to be revised or rejected.

Additionally, the conclusions reached regarding the effect of singularity distributions are specific to the implementation of the singularity distributions presented herein. As has already been pointed out, it would be possible to formulate the singularity distributions differently and thus obtain different behavior. Improving the singularity distributions is actually highly recommended, as will be discussed in the next section.

The placement of control points (described in Chapter 3) is also specific to the implementation considered here. It may be possible to place control points using a different method and thereby obtain different results.

Finally, for the sake of brevity, this work has been limited in scope to considering only Dirichlet boundary condition formulations. Based on the two Dirichlet formulations considered, certain conclusions were reached. However, it is possible to employ Neumann boundary condition formulations or even other Dirichlet formulations. It is anticipated that these different formulations would produce different results.

10.6 Future Work

There are still many theoretical aspects of panel methods left to explore, particularly for supersonic panel methods. In addition, many improvements can still be made to the pilot code developed as part of this work.

10.6.1 Improved Implementation of Higher-Order Singularity Distributions

The most readily-made and useful improvement to the panel method presented here would be to improve the higher-order singularity distributions. As shown in Chapters 8 and 9, the higher-order singularity distributions sometimes produced very poor results. These poor results seemed to be mostly tied to issues in the mesh, but the meshes did not cause poor results when the lower-order distributions were used. Thus, further work into improving the higher-order distributions may be valuable.

The comparison with PAN AIR (Section 9.5.1) was particularly insightful for seeing how these higher-order singularity distributions could be improved, as the results from PAN AIR were very good. For one, the higher-order distributions implemented in PAN AIR are weighted for supersonic flow such that any upstream influences are minimized [2, 33]. In addition, PAN AIR takes advantage of the regularity of a structured mesh to fit the doublet distribution based on evenly-distributed doublet strength locations. The midpoint-interpolation method used in MachLine would perform very well if the mesh were regular. However, this cannot be expected to be the case. Thus, it would be valuable to update the

midpoint-fit method to locate more points upon which to base the doublet distribution as well as ensure these points are well-distributed.

In doing so, the method of Lyu et al. [26] for determining cell potential distributions may be useful. In their work, Lyu et al. used an over-determined system of equations weighted by the distance of each known point from the cell of interest. The fact that the system was over-determined allowed for weighting the least-squares fit. This would also be useful as a weighted least-squares fit is what is used in PAN AIR for minimizing upstream influences in supersonic flow.

It was also observed that the higher-order singularity distributions tended to produce poor results over high aspect ratio panels. Due to the arbitrary nature with which unstructured meshes may be defined, it is likely that any given mesh could contain high aspect ratio panels. Thus, this aspect needs to be improved perhaps by again weighting doublet strength contributions by the inverse of their distance from the panel under consideration.

In addition, the results at configuration trailing edges were often found to be in error when higher-order distributions were used (see Sections 9.2 and 9.4). This may have been due to how trailing edges were marked as discontinuous and thus set to have a linear distribution of doublet strength along their length. This could be relaxed by allowing for quadratic doublet distributions on wake panels (discussed subsequently). This would also require finding a different method for specifying the midpoint doublet strengths on trailing edges. The requirement that the doublet strength not vary in the streamwise direction on the wake could prove useful for determining the trailing-edge midpoint strength. Alternatively, the doublet strength distribution on such panels could be determined in part by recognizing that the doublet distribution should satisfy Laplace's equation. This would provide an extra constraint equation to replace the effect of the missing neighboring doublet strength.

Also, the higher-order singularity distributions should be modified to be less sensitive to sudden qualitative shifts in the mesh (such as were seen in Sections 9.2 and 9.4). This would likely require using more than four neighboring points to determine a midpoint doublet

strength, or searching for neighboring points closer to the midpoint than the points originally used. This would fit well with a method modeled off of Lyu et al.'s approach [26].

With all these issues, it is worth asking whether the higher-order distributions are really worth using. As shown in Sections 8.2.3 and 8.2.4, the use of higher-order singularity distributions can improve the order of grid convergence over lower-order distributions for certain configurations and with the maximum continuity angle properly set. And as shown in Section 9.5, the results obtained when using higher-order distributions can be smoother and more accurate than when lower-order distributions are used, particularly for flat surfaces with varying flow properties. Thus, the higher-order singularity distributions presented here can be useful if continuity between panels is carefully controlled. Additionally, higher-order distributions may prove useful as part of an adaptive p -refinement scheme (discussed below).

10.6.2 Improvements to the FQRUP Solver

It has been shown here that a novel matrix solver, called the FQRUP solver, has the potential to compete with standard matrix solvers, such as GMRES, in terms of solution speed and accuracy for a supersonic case. However, there are drawbacks to the FQRUP solver. The main drawback is that the computed solution often resulted in a residual vector with a norm of 10^{-10} or greater. Compare this to standard LU decomposition, which typically produced a residual norm of less than 10^{-14} . While this performance from the FQRUP solver is good, it could be improved, likely by incorporating some form of pivoting.

Additionally, many matrix solvers (the block Jacobi (BJAC) solver is a good example) may be parallelized to improve performance. Implementing the FQRUP solver in parallel has not been explored here. However, it is anticipated that doing so could yield significant performance gains.

10.6.3 Superinclined Panels

The difficulties surrounding superinclined panels were described in Section 2.2. A clear and verified description of the correct method for calculating influence coefficients for superinclined panels is still needed. An influence coefficient calculation procedure has been given here (see Section 5.2) that matches the potential influence of a distribution of discrete doublets on a superinclined panel. However, due to the other difficulties with superinclined panels, and the lack of velocity influence calculations, these calculations have not been validated as part of the pilot code.

In addition, a correct understanding of the necessary boundary conditions on superinclined panels is needed. The boundary conditions used by others have been reviewed here but have not been implemented.

10.6.4 Improved Kutta Condition

Within MachLine, enforcement of the Kutta condition is very rudimentary. The flow is forced to be tangent to both the top and bottom surfaces of a wake-shedding edge, and this is assumed to cause the flow to leave the trailing edge smoothly. However, this is not a true reflection of the Kutta condition. In subsonic flow, the Kutta condition is usually expressed as a requirement that the velocity be finite at sharp trailing edges [23] or that the fluid pressure on either side of the trailing edge must be the same [87]. Either of these yield a nonlinear condition within a panel method, which must be solved iteratively [87]. It is to avoid this added complexity that the simplified Kutta condition is more often used within panel methods. However, an iterative method enforcing the true Kutta condition could improve accuracy for some cases.

In supersonic flow, the Kutta condition is different. For wings with supersonic trailing edges, the flows over the upper and lower surfaces are independent of one another. Thus, enforcing equal pressures would be inappropriate. For this case, the implementation provided in this work suffices. However, for wings with subsonic trailing edges, there is some interaction between upper and lower surfaces. How this relates to the enforcement of some-

thing similar to a Kutta condition is yet to be seen. It would be valuable to explore this aspect of supersonic panel methods.

10.6.5 Higher-Order Wake Panels

Along with an improved Kutta condition, higher-order panels could be implemented differently on wakes. The current implementation simply has wakes with linear doublet distributions since the method for handling discontinuous edges requires a linear distribution of doublet strength along wake-shedding edges. However, if another condition could be introduced at wake-shedding edges (since the doublet strength does not need to be continuous there), then the doublet strength along wake-shedding edges could be quadratic, leading to quadratic wake panels. This alternate condition would be necessary to fix the doublet strength at the midpoint of wake-shedding edges. This condition could come from a better approach to enforcing the Kutta condition, a fit of neighboring strengths, or something else entirely.

10.6.6 Wake Relaxation

As currently implemented, the pilot code simply models wake sheets as being flat and aligned with the freestream. However, the wake – being force-free – is convected along with the flow. This means the wake should be everywhere tangent to the local velocity [146]. It is common for subsonic panel methods to iteratively relax the shape of the wake such that this condition is met. However, this has been minimally explored for supersonic panel methods. It is anticipated that wake relaxation in supersonic flow poses unique challenges compared to subsonic flow, mostly due to the presence of shocks. To implement wake relaxation, it would be necessary to determine and implement the correct velocity influence formulas for supersonic flows.

10.6.7 Wake-Fuselage Stitching

In addition to wake relaxation, correctly modeling wake-fuselage interactions is critical for obtaining accurate results for configurations where such interactions would occur. Currently, wake-fuselage interactions are not handled in the pilot code, nor were these a focus of the current research effort. However, some comments can be made as to how these could be handled.

First, it is important to remember that the distributions of doublet strength must be continuous in supersonic flow. Because of this, it is suspected that implementing strand-based wakes as part of a supersonic panel code would be very difficult. Instead, panel-based wakes, such as currently implemented in the pilot code, would likely need to be expanded. VORLAX [30] is a vortex-lattice method that uses constant-strength doublet panels for modeling supersonic flow and may be a useful resource for those wishing to implement strand-based wakes as part of a supersonic panel method.

Second, continuing the use of Dirichlet boundary conditions would require allowing the wake to create a jump in potential on the surface of the fuselage. Wakes are sheets of vorticity (i.e. doublet sheets) that include a jump in potential from one side of the sheet to the other. If the wake trails aft of a lifting surface along the fuselage, then the surface potential on these panels must have a jump corresponding to the local wake strength. Moving to a Neumann formulation would alleviate this issue, though other difficulties would certainly be introduced.

Finally, the wake sheet must be relaxed and stitched to the fuselage in order to properly model interactions between the wake and the fuselage. This includes making the edge of the wake stay attached to and follow the fuselage downstream of lifting surfaces, as well as allowing the wake to split and go around aft lifting surfaces, if these are encountered. There is nothing theoretically difficult about these improvements. It is the implementation which is lacking.

10.6.8 Adaptive p -Refinement

In CFD and finite-element methods, it is often common to provide mesh refinement by increasing the order of the polynomials approximating the fluid unknowns within certain mesh cells. Doing so is called p -refinement. For example, if the unknowns are originally assumed to be linearly distributed in each mesh cell, then the polynomials can be made quadratic wherever greater resolution is needed and the simulation rerun. This type of refinement is opposed to splitting or shrinking mesh cells in that region to obtain the resolution needed.

As described in this work, nothing about the unknown singularity distributions requires the order of distribution to be the same between panels. One panel may have a linear-doublet-constant-source distribution while its neighbor has a quadratic-doublet-linear-source distribution. Thus, there is a possibility for implementing p -refinement in the supersonic panel method. How easy this would be to implement and the impact it would have on results is unknown. Also, metrics for automatically applied p -refinement would need to be developed. However, it has been shown here how higher-order distributions can produce better results than lower-order distributions for configurations with large flat surfaces that have varying flow properties (e.g. see Sections 8.2.3, 8.2.4, and 9.5). Thus, implementing p -refinement could prove useful

10.6.9 Neumann Boundary Conditions

In this work, only Dirichlet (potential-based) boundary conditions have been considered. However, Neumann (velocity-based) boundary conditions are possible and have been implemented in the past. Particularly since the Dirichlet boundary conditions enforce zero linearized mass flux, Neumann boundary conditions may be valuable for obtaining more accurate results for certain configurations (see [50,121]). Implementing them as part of the pilot code would mostly require pulling the necessary formulas from the relevant literature.

10.6.10 Multipole Methods for Influence Calculations

When a panel or group of panels is far from a control point, it is possible to calculate the influence of the panel(s) approximately using multipole expansions. Doing so sacrifices a small amount of accuracy in exchange for a significant improvement in computational speed. Similar approximations were implemented in PAN AIR [33,65]. Implementing these in MachLine could produce significant computational savings when assembling the linear system of equations.

REFERENCES

- [1] Anda, A. A. and Park, H., “Self-Scaling Fast Rotations for Stiff and Equality-Constrained Linear Least Squares Problems,” *Linear Algebra and its Applications*, Vol. 234, 1996, pp. 137–161.
- [2] Ehlers, F. E., Epton, M. A., Johnson, F. T., Magnus, A. E., and Rubbert, P. E., “A Higher Order Panel Method for Linearized Supersonic Flow,” NASA Contractor Report 3062, 1979.
- [3] Tamada, K., “On the Flow of a Compressible Fluid past a Sphere,” *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, Vol. 21, No. 12, 1939, pp. 743–752.
- [4] Schmitt, V. and Charpin, F., “Pressure Distributions on the Onera-M6-Wing at Transonic Mach Numbers,” *Experimental Data Base for Computer Program Assessment*, Vol. 138, Advisory Group for Aerospace Research and Development, 1979, pp. 335–337, 364.
- [5] National Aeronautics and Space Administration, “Tables for Supersonic Flow Around Right Circular Cones at Zero Angle of Attack,” NASA SP-3004, 1964.
- [6] Love, E. S., “Investigations at Supersonic Speeds of 22 Triangular Wings Representing Two Airfoil Sections for Each of 11 Apex Angles,” NACA Research Memorandum No. L9D07, 1955.
- [7] Davis, J. D., *A Higher-Order Method Implemented in an Unstructured Panel Code to Model Linearized Supersonic Flows*, California Polytechnic State University, San Luis Obispo, 2019.
- [8] Hayes, W. D., Browne, S. H., and Lew, R. J., “Linearized Theory of Conical Supersonic Flow With Application to Triangular Wings,” Report No. NA-46-818, North American Aviation, Inc., 1946.
- [9] Smith, A. M. O., “The Panel Method: Its Original Development,” *Applied Computational Aerodynamics*, AIAA, Washington, DC, 1989, pp. 3–17.
- [10] Choi, S., Alonso, J. J., and Kroo, I. M., “Two-Level Multi-Fidelity Design Optimization Studies for Supersonic Jets,” *Journal of Aircraft*, Vol. 46, 2009.
- [11] Choi, S., Alonso, J. J., Kroo, I. M., and Wintzer, M., “Multifidelity Design Optimization of Low-Boom Supersonic Jets,” *Journal of Aircraft*, Vol. 45, 2008.
- [12] Allison, D. L., Morris, C. C., Schetz, J. A., Kapania, R. K., Watson, L. T., and Deaton, J. D., “Development of a multidisciplinary design optimization framework for an efficient supersonic air vehicle,” *Advances in Aircraft and Spacecraft Science*, Vol. 2, 2015, pp. 17–44.

- [13] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "High-Fidelity Aerostructural Design Optimization of a Supersonic Business Jet," *Journal of Aircraft*, Vol. 41, 2004.
- [14] Kroo, I., Willcox, K., March, A., Haas, A., Rajnarayan, D., and Kays, C., "Multi-fidelity Analysis and Optimization for Supersonic Design," NASA CR-2010/216874, 2010.
- [15] Alonso, J. J. and Kroo, I. M., "Advanced Algorithms for Design and Optimization of Quiet Supersonic Platforms," *40th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Reno, NV, 2002.
- [16] Chen, P. C., Liu, D. D., Chang, K. T., Gao, X. W., and Tang, L., "Hypersonic Aerothermodynamics Methodology using ZONAIR for RLV/TPS Design and Analysis," *14th Thermal Fluids Analysis Workshop*, NASA, Hampton, VA, 2003.
- [17] Demiroglu, Y., Yildiz, S., and Nikba, M., "Multi-fidelity Sonic Boom Minimization of a Supersonic Aircraft by Parametric Wing Shape Design," *SciTech Forum*, AIAA, Virtual Event, 2021.
- [18] Chan, M. K., *Supersonic Aircraft Optimization for Minimizing Drag and Sonic Boom*, Stanford University, Stanford, August 2003.
- [19] Giblette, T. N., *Rapid Prediction of Low-Boom and Aerodynamic Performance of Supersonic Transport Aircraft Using Panel Methods*, Utah State University, 2019.
- [20] Glauert, H., "The Effect of Compressibility on the Lift of an Airfoil," *Proceedings of the Royal Society of London*, Vol. 118, 1928.
- [21] Ward, G. N., *Linearized Theory of Steady High-Speed Flow*, Cambridge University Press, Great Britain, 1955.
- [22] Mase, G. T., Smelser, R. E., and Mase, G. E., *Continuum Mechanics for Engineers*, CRC Press, Boca Raton, FL, 3rd ed., 2010.
- [23] Thwaites, B., *Incompressible Aerodynamics*, Oxford at the Clarendon Press, Oxford, 1960.
- [24] Rotty, R. M., *Introduction to Gas Dynamics*, John Wiley & Sons, Inc., New York, 1962.
- [25] Morino, L. and Iemma, U., "Boundary integral equations and conservative dissipation schemes for full potential transonic flows," *Computational Mechanics*, Vol. 13, 1993, pp. 90–99.
- [26] Lyu, F., Xiao, T., and Yu, X., "A fast and automatic full-potential finite volume solver on Cartesian grids for unconventional configurations," *Chinese Journal of Aeronautics*, Vol. 30, 2017, pp. 951–963.
- [27] Sinclair, P. M., "A three-dimensional field-integral method for the calculation of transonic flow on complex configurations - theory and preliminary results," *The Aeronautical Journal*, Vol. 92, 1988, pp. 235–241.

- [28] Clever, W. C. and Shankar, V., “Nonlinear Potential Analysis Techniques for Supersonic/Hypersonic Configuration Design,” NASA Contractor Report 166078, 1983.
- [29] Robinson, A., “On Source and Vortex Distributions in the Linearized Theory of Steady Supersonic Flow,” *Quarterly Journal of Mechanics and Applied Mathematics*, Vol. 1, 1948, pp. 408–432.
- [30] Miranda, L. R., Elliott, R. D., and Baker, W. M., “A Generalized Vortex Lattice Method for Subsonic and Supersonic Flow Applications,” NASA Contractor Report 2865, 1977.
- [31] Ramm, H. J., *Fluid Dynamics for the Study of Transonic Flow*, Oxford University Press, New York, 1990.
- [32] Sinclair, P. M., “An Exact Integral (Field Panel) Method for the Calculation of Two-Dimensional Transonic Potential Flow around Complex Configurations,” *The Aeronautical Journal*, Vol. 90, 1986, pp. 227–236.
- [33] Epton, M. A. and Magnus, A. E., “PAN AIR: A computer program for predicting subsonic or supersonic linear potential flows about arbitrary configurations using a higher order panel method. Volume 1: Theory document.” NACA-CR-3251, November 1981.
- [34] Anderson, J. D., *Modern Compressible Flow With Historical Perspective*, McGraw-Hill, 3rd ed., 2003.
- [35] Salas, M. D. and Iollo, A., “Entropy Jump Across an Inviscid Shock Wave,” *Theoretical and Computational Fluid Dynamics*, Vol. 8, 1996, pp. 365–375.
- [36] Hafez, M., “Some Computational Problems of Transonic Aerodynamics,” *39th Aerospace Sciences Meeting*, AIAA, Reno, NV, 1999.
- [37] Ballhaus, W. F. and Goorjian, P. M., “Implicit Finite-Difference Computations of Unsteady Transonic Flows about Airfoils,” *AIAA Journal*, Vol. 15, 1977.
- [38] Piers, W. J. and Slooff, J. W., “Calculation of Transonic Flow by Means of a Shock-Capturing Field Panel Method,” *4th Computational Fluid Dynamics Conference*, AIAA, 1979, pp. 147–156.
- [39] Prince, S. A., di Pasquale, D., and Garry, K., “Progress Towards a Rapid Method for Conceptual Aerodynamic Design for Transonic Cruise,” *AIAA SciTech Forum*, AIAA, Orlando, FL, 2020.
- [40] Hunt, B. and Adamson, A. P., “Knowledge-based shock-wave modeling by GENESIS,” *Applied Mechanics Reviews*, Vol. 46, 1993.
- [41] Strauss, W. A., *Partial Differential Equations: An Introduction*, John Wiley & Sons, Inc., Hoboken, NJ, 2nd ed., 2008.
- [42] Morse, P. M. and Feshbach, H., *Methods of Theoretical Physics: Part I*, McGraw-Hill Book Company, Inc., New York, NY, 1953.

- [43] Kellogg, O. D., *Foundations of Potential Theory*, Dover Publications, Inc., 1953.
- [44] Brebbia, C. A., *The Boundary Element Method for Engineers*, John Wiley & Sons, Inc., New York, 1978.
- [45] Johnson, F. T., "A General Panel Method for the Analysis and Design of Arbitrary Configurations in Incompressible Flows," NASA Contractor Report 3079, 1980.
- [46] Katz, J. and Plotkin, A., *Low-Speed Aerodynamics*, Cambridge University Press, Cambridge, UK, 2nd ed., 2001.
- [47] Morino, L., "Boundary Integral Equations in Aerodynamics," *Applied Mechanics Reviews*, Vol. 46, 1993.
- [48] Anderson Jr, J. D., *Fundamentals of Aerodynamics*, McGraw-Hill, 5th ed., 2011.
- [49] Maskew, B., "Program VSAERO Theory Document: A Computer Program for Calculating Nonlinear Aerodynamic Characteristics of Arbitrary Configurations," NASA Contractor Report 4023, 1987.
- [50] Erickson, L. L., "Panel Methods - An Introduction," NASA Technical Paper 2995, 1990.
- [51] Beer, G., *Programming the Boundary Element Method: An Introduction for Engineers*, Wiley, 2001.
- [52] Hess, J. L., "Panel Methods in Computational Fluid Dynamics," *Annual Review of Fluid Mechanics*, Vol. 22, 1990, pp. 255–274.
- [53] Hess, J. L. and Smith, A. M. O., "Calculation of Potential Flow about Arbitrary Bodies," *Progress in Aerospace Sciences*, Vol. 8, 1967, pp. 1–138.
- [54] Miranda, L., "Application of Computational Aerodynamics to Aircraft Design," *Journal of Aircraft*, Vol. 21, 1984.
- [55] Woodward, F. A., "Analysis and Design of Wing-Body Combinations at Subsonic and Supersonic Speeds," *Journal of Aircraft*, Vol. 5, 1968, pp. 528–534.
- [56] Carmichael, R. L. and Woodward, F. A., "An Integrated Approach to the Analysis and Design of Wings and Wing-Body Combinations in Supersonic Flow," NASA Technical Note D-3685, 1966.
- [57] Woodward, F. A., "USSAERO Computer Program Development, Versions B and C," NACA Contractor Report 3227, 1980.
- [58] Woodward, F. A., "An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic and Supersonic Flow Part I - Theory and Application," NACA Contractor Report 2228 Part I, 1973.
- [59] Fornasier, L., "Treatment of Supersonic Configurations by an Updated Low-Order Panel Method," *Journal of Aircraft*, Vol. 21, 1984, pp. 227–237.

- [60] Tinoco, E. N. and Rubbert, P. E., "Panel Methods: PAN AIR," *Computational Methods in Potential Aerodynamics*, edited by L. Morino, Springer-Verlag, New York, 1985, pp. 39–93.
- [61] Fornaseir, L., "HISSS - A Higher-Order Subsonic/Supersonic Singularity Method for Calculating Linearized Potential Flow," *17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference*, AIAA, Snowmass, CO, 1984, p. 17.
- [62] Miranda, L. R., "CFD Applications: The Lockheed Perspective," *Supercomputing in Aerospace*, NASA Ames Research Center, 1987, pp. 77–85.
- [63] Youngren, H. H., Bouchard, E. E., Coopersmith, R. M., and Miranda, L. R., "Comparison of Panel Method Formulations and its Influence on the Development of QUAD-PAN, an Advanced Low Order Panel Method," *Applied Aerodynamics Conference*, AIAA, Danvers, MA, 1983, p. 14.
- [64] Morino, L., "Steady, Oscillatory, and Unsteady Subsonic and Supersonic Aerodynamics - Production Version (SOUSSA-P.1,1), Vol. I, Theoretical Manual," NASA Contractor Report 159130, 1980.
- [65] PDAS, "PAN AIR," Online <https://www.pdas.com/panair.html>, 2021.
- [66] Maruyama, Y., Akishita, S., and Nakamura, A., "New panel method for supersonic flows about arbitrary configurations," *Journal of Aircraft*, Vol. 25, 1988.
- [67] Maruyama, Y., Akishita, S., and Nakamura, A., "Numerical Simulations of Supersonic Flows about Arbitrary Configurations Using a New Panel Method Program," *Aerodynamics Conference*, AIAA, Williamsburg, VA, 1986, pp. 303–312.
- [68] Davis, J. D. and Marshall, D. D., "A Higher-Order Method Implemented in an Unstructured Panel Code to Model Linearized Supersonic Flows," *SciTech Forum*, AIAA, Orlando, FL, 2020.
- [69] Hoeijmakers, H. W. M., "A Panel Method for the Determination of the Aerodynamic Characteristics for Complex Configurations in Linearized Subsonic or Supersonic Flow," Tech. Rep. 80124, NLR, 1980.
- [70] Crowley, J. A., Marland, D., and Marchbank, W. R., "Development of a Supersonic Panel Program. Phase II: Development within the Warton Mk II CFM System, of Version I of a Lifting Supersonic Panel Program (Sup PP(VI))," Tech. rep., BAe Warton, 1984.
- [71] Smith, J. S. and Woodward, D. S., "An Assessment of the Use of Low-Order Panel Methods for the Calculation of Supersonic Flows," *The Aeronautical Journal*, Vol. 91, 1987, pp. 314–320.
- [72] Ahuja, V., *Aerodynamic Loads over Arbitrary Bodies by Method of Integrated Circulation*, Auburn University, Auburn, August 2013.

- [73] Satterwhite, C. R., *Development of CPanel, an Unstructured Panel Code, Using a Modified TLS Velocity Formulation*, California Polytechnic State University, San Luis Obispo, 2015.
- [74] Sousa, C., *Unsteady Panel Code Utilizing a Vortex Particle Wake*, California Polytechnic State University, San Luis Obispo, 2016.
- [75] Morino, L., Bernardini, G., and Gennaretti, M., “A Velocity-Potential-Based Boundary-Element Method for the Aeroacoustic Analysis of Rotors and Propellers in Arbitrary Motion,” *8th AIAA/CEAS Aeroacoustics Conference & Exhibit*, AIAA, 2002.
- [76] Conlan-Smith, C., Ramos-Garcia, N., Sigmund, O., and Andreasen, C. S., “Aerodynamic Shape Optimization of Aircraft Wings Using Panel Methods,” *AIAA Journal*, Vol. 58, 2020.
- [77] Quijada, G. M. and Boschetti, P. J., “Linear Computational Fluid Dynamic Analysis of Dynamic Ground Effect of a Wing in Sink and Flare Maneuvers,” *SciTech Forum*, AIAA, Kissimmee, FL, 2015.
- [78] Hunt, B., “GENESIS A Mesh-Free, Knowledge-Based Nonlinear Boundary Integral Methodology for Compressible, Viscous Flows over Arbitrary Bodies: Theoretical Framework and Basic Physical Principles,” *Proceedings of the IABEM Symposium*, Rome, 1990, pp. 241–250.
- [79] Johnson, F. T. and Rubbert, P., “Advanced Panel-Type Influence Coefficients Methods Applied to Subsonic Flows,” *13th Aerospace Sciences Meeting*, AIAA, Pasadena, CA, 1975, p. 10.
- [80] Maskew, B., “Prediction of Subsonic Aerodynamic Characteristics - A Case for Low-Order Panel Methods,” *Journal of Aircraft*, Vol. 19, 1982, pp. 157–163.
- [81] Chen, P. C. and Liu, D. D., “Unified Hypersonic/Supersonic Panel method for Aeroelastic Applications to Arbitrary Bodies,” *Journal of Aircraft*, Vol. 39, 2002.
- [82] ZONA Technology Inc., *ZONAIR Version 4.5 User’s Manual*, ZONA Technology Inc., Scottsdale, AZ, 2014.
- [83] Sun, Y. and Smith, H., “Sonic Boom and Drag Evaluation of Supersonic Jet Concepts,” *Aeroacoustics Conference*, AIAA/CEAS, Atlanta, GA, 2018.
- [84] Hirschel, E. H. and Fornasier, L., “Flowfield and Vorticity Distribution Near Wing Trailing Edges,” *22nd Aerospace Sciences Meeting*, AIAA, Reno, NV, 1984, p. 9.
- [85] Mercer, J. E., Weber, J. A., and Lesferd, E. P., “Aerodynamic Influence Coefficient Method using Singularity Splines,” NASA Contractor Report CR-2423, 1974.
- [86] Rubbert, P. E., “Sideslip of Wing-Body Combinations,” NASA Contractor Report 114716, 1972.

- [87] Hess, J. L., Friedman, D. M., and Clark, R. W., "Calculation of Compressible Flow about Three-Dimensional Inlets with Auxiliary Inlets, Slats and Vanes by Means of a Panel Method," NACA Contractor Report 174975, 1985.
- [88] Gopal, V. and Maddalena, L., "Influence of vorticity distribution on singularities in linearized supersonic flow," *Physics of Fluids*, Vol. 30, 2018.
- [89] Cenko, A., Tinoco, E. N., Dyer, R. D., and DeJongh, J., "PAN AIR Applications to Weapons Carriage and Separation," *Journal of Aircraft*, 1981, pp. 128–134.
- [90] Madson, M. D. and Erickson, L. L., "PAN AIR Analysis of the NASA/MCAIR 279-3: An Advanced Supersonic V/STOL Fighter/Attack Aircraft," NASA Technical Memorandum 86838, 1986.
- [91] Tinoco, E. N. and Johnson, F. T., "Application of a Higher Order Panel Method to Realistic Supersonic Configurations," *Journal of Aircraft*, Vol. 17, 1980.
- [92] Herling, W. W., Holcomb, J. E., Saheli, F. P., and Hansen, R. L., "Prediction of the Steady Three-Dimensional Aerodynamics of a Supersonic Ramjet Missile," *21st Joint Propulsion Conference*, AIAA, Monterey, 1985, p. 9.
- [93] Rajnarayan, D., Haas, A., and Kroo, I., "A Multifidelity Gradient-Free Optimization Method and Application to Aerodynamic Design," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, 2008.
- [94] Liu, D. D., Chen, P. C., Tang, L., and Chang, K. T., "Hypersonic Aerothermodynamics/Aerothermoelastics Methodology for Reusable Launch Vehicles/TPS Design and Analysis," *41st Aerospace Sciences Meeting and Exhibit*, AIAA, Reno, NV, 2003.
- [95] Mansur, W. J., Araujo, F. C., and Malaghini, J. E. B., "Solution of BEM Systems of Equations via Iterative Techniques," *International Journal for Numerical Methods in Engineering*, Vol. 33, 1992, pp. 1823–1841.
- [96] Davey, K. and Bounds, S., "A Generalized SOR Method for Dense Linear Systems of Boundary Element Equations," *SIAM Journal of Scientific Computing*, Vol. 19, 1998, pp. 935–967.
- [97] Mullen, R. L. and Rencis, J. J., "Iterative Methods for Solving Boundary Element Equations," *Computers and Structures*, Vol. 25, 1987, pp. 713–723.
- [98] Chen, K., *Matrix Preconditioning Techniques and Applications*, Cambridge University Press, Cambridge, 2005.
- [99] Barra, L. P. S., Coutinho, A. L. G. A., Mansur, W. J., and Telles, J. C. F., "Iterative Solution of BEM Equations by GMRES Algorithm," *Computers & Structures*, Vol. 44, 1992, pp. 1249–1253.
- [100] Xiao, H. and Chen, Z., "Numerical Experiments of Preconditioned Krylov Subspace Methods Solving the Dense Non-Symmetric Systems Arising from BEM," *Engineering Analysis with Boundary Elements*, Vol. 31, 2007, pp. 1013–1023.

- [101] Boschitsch, A., Curbishley, T., Quackenbush, T., and Teske, M., "A fast panel method for potential flows about complex geometries," *34th Aerospace Sciences Meeting and Exhibit*, AIAA, Reno, NV, 1996.
- [102] Willis, D. J., Peraire, J., and White, J. K., "A combined pFFT-multipole tree code, unsteady panel method with vortex particle wakes," *International Journal for Numerical Methods in Fluids*, Vol. 53, 2007, pp. 1399–1422.
- [103] Yao, Z., Zheng, X., Yuan, H., and Feng, J., "Research progress of high-performance BEM and investigation on convergence of GMRES in local stress analysis of slender real thin-plate beams," *Engineering Computations*, Vol. 36, No. 8, Oct. 2019, pp. 2530–2556.
- [104] Sun, J., Zheng, X., Liu, Y., and Yao, Z., "Some investigations on convergence of GMRES in solving BEM equations for slender beam structures," *Engineering Analysis with Boundary Elements*, Vol. 126, 2021, pp. 128–135.
- [105] Chen, P. C., Gao, X. W., and Tang, L., "Overset Field-Panel Method for Unsteady Transonic Aerodynamic Influence Coefficient Matrix Generation," *AIAA Journal*, Vol. 42, 2004.
- [106] Deconinck, H. and Hirsch, C., "Transonic Flow Calculations with Higher Order Finite Elements," *Seventh International Conference on Numerical Methods in Fluid Dynamics*, Stanford, CA, 1980, pp. 138–143.
- [107] Johnson, F. T., James, R. M., Bussoletti, J. E., Woo, A. C., and Young, D. P., "A Transonic Rectangular Grid Embedded Panel Method," *3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference*, AIAA/ASME, St. Louis, MO, 1982, p. 20.
- [108] Bussoletti, J. E., Johnson, F. T., Young, D. P., Melvin, R. G., Burkhart, R. H., Bieterman, M. B., Samant, S. S., and SenGupta, G., "TRANAIR Technology: Solutions for Large PDE Problems," *Solution of Superlarge Problems in Computational Mechanics*, edited by J. H. Kane, A. D. Carlson, and D. L. Cox, Plenum Press, New York, NY, 1989, pp. 95–124.
- [109] Rubbert, P. E., Bussoletti, J. E., Johnson, F. T., Sidwell, K. W., Rowe, W. S., Samant, S. S., SanGupta, G., Weatherill, W. H., Burkhart, R. H., Everson, B. L., Young, D. P., and Woo, A. C., "A New Approach to the Solution of Boundary Value Problems Involving Complex Configurations," Vol. 75, 1986, pp. 49–84.
- [110] Samant, S. S., Bussoletti, J. E., Johnson, F. T., Burkhart, R. H., Everson, B. L., Melvin, R. G., Young, D. P., Erickson, L. L., Madson, M. D., and Woo, A. C., "TRANAIR: A Computer Code for Transonic Analyses of Arbitrary Configurations," *25th Aerospace Sciences Meeting*, AIAA, Reno, NV, 1987, p. 18.
- [111] Hunt, B., "Generalized nonlinear boundary integral methods for compressible, viscous flows over arbitrary bodies: knowledge-based CFD," *Engineering Analysis with Boundary Elements*, Vol. 12, 1993, pp. 163–174.

- [112] Hunt, B. and Adamson, A. P., “Current Status of the GENESIS Methodology for Knowledge-Based Treatment of Transonic Flows, with Emphasis on Shock Fitting and Non-Dissipative Suppression of Expansion Shocks,” *Proceedings of the IABEM Symposium*, Kyoto, 1991, pp. 111–128.
- [113] Hunt, B. and Plybon, R. C., “Generalization of the Boundary Integral Method to Nonlinear Problems of Compressible Fluid Flow: The No-mesh Alternative - Part I,” *Proceedings of the International Symposium on Boundary Element Methods: Advances in Solid and Fluid Mechanics*, East Hartford, CN, 1989, pp. 147–154.
- [114] Hunt, B. and Plybon, R. C., “Generalization of the Boundary Integral Method to Nonlinear Problems of Compressible Fluid Flow: The No-mesh Alternative - Part II,” *Proceedings of the International Symposium on Boundary Element Methods: Advances in Solid and Fluid Mechanics*, East Hartford, CN, 1989, pp. 155–159.
- [115] Ramos, A., *Development of a Meshless Method to Solve Compressible Potential Flows*, California Polytechnic State University, San Luis Obispo, CA, 2010.
- [116] Crovato, A., Almeida, H. S., Vio, G., and Silva, G. H., “Effect of Levels of Fidelity on Steady Aerodynamic and Static Aeroelastic Computations,” *Aerospace*, Vol. 7, 2020.
- [117] Goates, C. D. and Hunsaker, D. F., “Development of a Subsonic-Supersonic, Unstructured Panel Method,” *SciTech Forum*, AIAA, San Diego, CA, 2022.
- [118] AeroLab, “MachLine,” <https://github.com/usuaero/MachLine>, 2022.
- [119] Goates, C. D., “MachLine: Dissertation Version,” code archive accessible at doi.org/10.5281/zenodo.8253561, 2023.
- [120] Morino, L., “A General Theory of Unsteady Compressible Potential Aerodynamics,” NASA Contractor Report 2464, 1974.
- [121] Towne, M., Strande, S., Erickson, L., Kroo, I., Enomoto, F., Carmichael, R., and McPherson, K., “PAN AIR Modeling Studies,” *Applied Aerodynamics Conference*, AIAA, Danvers, MA, 1983, p. 26.
- [122] Lyman, V. and Hancock, J., “Comparison of Theory and Experiment for Propfan Inlets,” *Journal of Propulsion and Power*, Vol. 3, 1987.
- [123] Ahuja, V. and Hartfield, R., “Aerodynamic Loads over Arbitrary Bodies by Method of Integrated Circulation,” *Journal of Aircraft*, Vol. 53, 2016, pp. 1719–1730.
- [124] Sidwell, K. W., Baruah, P. K., Bussioletti, J. E., Medan, R. T., Conner, R. S., and Purdon, D. J., “PAN AIR - A Computer Program for Predicting Subsonic or Supersonic Linear Potential Flows About Arbitrary Configurations Using a Higher Order Panel Method - Volume II - User’s Manual,” Tech. Rep. CR3252, NASA, 1981.
- [125] Goates, C. D., Houser, A. M., and Hunsaker, D. F., “Implementation of MachLine: A Subsonic/Supersonic, Unstructured Panel Code,” *SciTech Forum*, AIAA, National Harbor, MD, 2023.

- [126] Moon, T. K. and Stirling, W. C., *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall, 2000.
- [127] Hoeijmakers, H. W. M., “Panel methods in Aerodynamics; Some Highlights,” *Panel Methods in Fluid Mechanics with Emphasis on Aerodynamics*, 1987.
- [128] Ehrlich, L. W., “The Block Symmetrix Successive Overrelaxation Method,” *Journal of the Society for Industrial and Applied Mathematics*, Vol. 12, 1964, pp. 807–826.
- [129] Hageman, L. A. and Young, D. M., *Applied Iterative Methods*, Academic Press, New York, 1981.
- [130] Hobgen, L., editor, *Handbook of Linear Algebra*, Chapman & Hall/CRC, Boca Raton, FL, 2007.
- [131] Shakib, F., Hughes, T. J., and Johan, Z., “A multi-element group preconditioned GMRES algorithm for nonsymmetric systems arising in finite element analysis,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 75, No. 1-3, Oct. 1989, pp. 415–456.
- [132] Stewart, G. W., *Matrix Algorithms Volume 1: Basic Decompositions*, SIAM, Philadelphia, PA, 1998.
- [133] Gentleman, W. M., “Least Squares Computations by Givens Transformations Without Square Roots,” *Journal of the Institute of Mathematics and its Applications*, Vol. 12, 1973, pp. 329–336.
- [134] Golub, G. H. and Loan, C. F. V., *Matrix Computations*, Johns Hopkins University Press, 1996.
- [135] Anda, A. A. and Park, H., “Fast Plane Rotations with Dynamic Scaling,” *SIAM Journal on Matrix Analysis and Applications*, Vol. 15, 1994, pp. 162–174.
- [136] Reichel, L. and Ye, Q., “Breakdown-Free GMRES for Singular Systems,” *SIAM J. Matrix Anal. Appl.*, Vol. 26, 2005, pp. 1001–1021.
- [137] Baker, A. H., Jessup, E. R., and Manteuffel, T., “A Technique for Accelerating the Convergence of Restarted GMRES,” *SIAM J. Matrix Anal. Appl.*, Vol. 26, 2005, pp. 962–984.
- [138] Houser, A. M., Goates, C. D., and Hunsaker, D. F., “Evaluation of the MachLine Subsonic-Supersonic Panel Code Against Experimental Results,” *SciTech Forum*, AIAA, National Harbor, MD, 2023.
- [139] von Karman, T., “Compressibility Effects in Aerodynamics,” *Journal of the Aeronautical Sciences*, Vol. 8, Sept 1941.
- [140] Laitone, E. V., “New Compressibility Correction for Two-Dimensional Subsonic Flow,” *Journal of the Aeronautical Sciences*, Vol. 18, May 1951.
- [141] Anderson, J. D., *Fundamentals of Aerodynamics*, McGraw-Hill, 6th ed., 2017.

- [142] Tamada, K., “Further studies on the flow of a compressible fluid past a sphere,” *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, Vol. 22, No. 7, 1940, pp. 519–525.
- [143] Mayeur, J., Dumont, A., Destarac, D., and Gleize, V., “RANS Simulations on TMR 3D Test Cases with the Onera elsA Flow Solver,” *SciTech Forum*, AIAA, 2016.
- [144] Phillips, W. F., *Mechanics of Flight*, chap. 1.9, John Wiley & Sons, Inc., Hoboken, 2nd ed., 2010, pp. 94–107.
- [145] Davies, M. and Saverin, J., “Implementation of Adaptive Gaussian Quadrature for Improved Accuracy of Boundary Element Methods Applied to Three Dimensional Geometries,” *Turbomachinery Technical Conference and Exposition*, ASME, Phoenix, AZ, 2019.
- [146] Sipicic, S. R. and Morino, L., “Wake Dynamics for Incompressible and Compressible Flows,” *Computational Methods in Potential Aerodynamics*, edited by L. Morino, Springer-Verlag, Berlin, 1985, pp. 679–699.
- [147] Morino, L., “Mathematical Foundations of Integral-Equation Methods,” *Computational Methods in Potential Aerodynamics*, edited by L. Morino, Springer-Verlag, Berlin, 1985, pp. 269–291.
- [148] Brebbia, C. A. and Walker, S., *Boundary Element Techniques in Engineering*, Newnes-Butterworths, London, 1980.
- [149] Wylie, C. R., *Advanced Engineering Mathematics*, McGraw-Hill, New York, 2nd ed., 1960.

APPENDICES

APPENDIX A
 PROOF THAT THE INVERSE OF R_β OR R_B SATISFIES THE PRANDTL-GLAUERT
 EQUATION

Previously, it was asserted that

$$\nabla[B]\nabla\phi = 0 \quad (\text{A.1})$$

where

$$\phi = \frac{1}{R_\beta} \quad (\text{A.2})$$

for subsonic flows and

$$\phi = \frac{1}{R_B} \quad (\text{A.3})$$

for supersonic flows. Only the subsonic case will be considered here, as the two cases differ only in nomenclature for this proof. Recall

$$\beta^2 = 1 - M_\infty^2 \quad (\text{A.4})$$

and

$$R_\beta = [(x - \xi)^2 + \beta^2(y - \eta)^2 + \beta^2(z - \zeta)^2]^{\frac{1}{2}} \quad (\text{A.5})$$

Thus, ϕ may be written

$$\phi = [(x - \xi)^2 + \beta^2(y - \eta)^2 + \beta^2(z - \zeta)^2]^{-\frac{1}{2}} \quad (\text{A.6})$$

Taking the first derivative with respect to x yields

$$\frac{\partial \phi}{\partial x} = - [(x - \xi)^2 + \beta^2(y - \eta)^2 + \beta^2(z - \zeta)^2]^{-\frac{3}{2}} (x - \xi) \quad (\text{A.7})$$

The second derivative is then equal to

$$\begin{aligned} \frac{\partial^2 \phi}{\partial x^2} = & - [(x - \xi)^2 + \beta^2(y - \eta)^2 + \beta^2(z - \zeta)^2]^{-\frac{3}{2}} \\ & + 3 [(x - \xi)^2 + \beta^2(y - \eta)^2 + \beta^2(z - \zeta)^2]^{-\frac{5}{2}} (x - \xi)^2 \end{aligned} \quad (\text{A.8})$$

Taking the first derivative with respect to y yields

$$\frac{\partial \phi}{\partial y} = -\beta^2 [(x - \xi)^2 + \beta^2(y - \eta)^2 + \beta^2(z - \zeta)^2]^{-\frac{3}{2}} (y - \eta) \quad (\text{A.9})$$

The second derivative is then equal to

$$\begin{aligned} \frac{\partial^2 \phi}{\partial y^2} = & -\beta^2 [(x - \xi)^2 + \beta^2(y - \eta)^2 + \beta^2(z - \zeta)^2]^{-\frac{3}{2}} \\ & + 3\beta^4 [(x - \xi)^2 + \beta^2(y - \eta)^2 + \beta^2(z - \zeta)^2]^{-\frac{5}{2}} (y - \eta)^2 \end{aligned} \quad (\text{A.10})$$

The second derivative with respect to z is almost identical

$$\begin{aligned} \frac{\partial^2 \phi}{\partial z^2} = & -\beta^2 [(x - \xi)^2 + \beta^2(y - \eta)^2 + \beta^2(z - \zeta)^2]^{-\frac{3}{2}} \\ & + 3\beta^4 [(x - \xi)^2 + \beta^2(y - \eta)^2 + \beta^2(z - \zeta)^2]^{-\frac{5}{2}} (z - \zeta)^2 \end{aligned} \quad (\text{A.11})$$

Combining the above results in Eq. (A.1) results in

$$\begin{aligned}
\nabla[B]\nabla\phi = & -\beta^2 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{3}{2}} \\
& + 3\beta^2 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{5}{2}} (x-\xi)^2 \\
& - \beta^2 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{3}{2}} \\
& + 3\beta^4 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{5}{2}} (y-\eta)^2 \\
& - \beta^2 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{3}{2}} \\
& + 3\beta^4 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{5}{2}} (z-\zeta)^2
\end{aligned}$$

$$\begin{aligned}
\nabla[B]\nabla\phi = & -3\beta^2 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{3}{2}} \\
& + 3\beta^2 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{5}{2}} (x-\xi)^2 \\
& + 3\beta^4 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{5}{2}} (y-\eta)^2 \\
& + 3\beta^4 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{5}{2}} (z-\zeta)^2
\end{aligned}$$

$$\begin{aligned}
\nabla[B]\nabla\phi = & -3\beta^2 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{3}{2}} \\
& + 3\beta^2 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{5}{2}} [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]
\end{aligned}$$

$$\begin{aligned}
\nabla[B]\nabla\phi = & -3\beta^2 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{3}{2}} \\
& + 3\beta^2 [(x-\xi)^2 + \beta^2(y-\eta)^2 + \beta^2(z-\zeta)^2]^{-\frac{3}{2}}
\end{aligned}$$

$$\nabla[B]\nabla\phi = 0$$

Of course, this falls apart whenever (x, y, z) and (ξ, η, ζ) are coincident. Handling such a case requires using the theory of generalized functions and is outside the scope of this work.

APPENDIX B

ALTERNATE DERIVATION OF THE UNIFIED BOUNDARY INTEGRAL EQUATION

Previously, it was shown how the unified BIE (Eq. (1.122)) may be derived by deriving Green's third identity for both subsonic and supersonic flows. An alternative derivation of Eq. (1.122) exists that relies more heavily on the method of Green's functions and was preferred by Morino (e.g. see [25]). To apply the method of Green's functions to the flow of an inviscid, irrotational fluid, first consider a flow (defined by its perturbation potential ϕ) which satisfies the Prandtl-Glauert equation (Eq. (1.58)). The Green's function for this equation is the function G that satisfies the equation [25, 42, 147]

$$(1 - M_\infty^2) G_{xx} + G_{yy} + G_{zz} = \delta(\mathbf{r} - \mathbf{r}_\infty) \quad (\text{B.1})$$

where δ is the Dirac delta function. If $M_\infty > 1$, it can be shown that this equation is satisfied by [2, 25, 120, 148]

$$G = -\frac{H\left(x - \xi - B\sqrt{(y - \eta)^2 + (z - \zeta)^2}\right)}{2\pi R_B} \quad (\text{B.2})$$

where $H()$ is the Heaviside step function, and R_B is the hyperbolic distance given by

$$R_B = \sqrt{(x - \xi)^2 + (1 - M_\infty^2)[(y - \eta)^2 + (z - \zeta)^2]} \quad (\text{B.3})$$

For subsonic flows, the Green's function is $G = -1/(4\pi R_B)$ (the Heaviside function is not needed in this case since the governing equation is elliptic). The Green's function for the Prandtl-Glauert equation represents a supersonic (or subsonic) source of unit strength [2].

This fundamental solution may be applied to an arbitrary flow volume V bounded by a surface S using a form of Green's second identity [43], here given as [21]

$$\iint_S (\phi_1 \mathbf{w}_2 - \phi_2 \mathbf{w}_1) \cdot \mathbf{n} dS = \iiint_V (\phi_1 \nabla \cdot \mathbf{w}_2 - \phi_2 \nabla \cdot \mathbf{w}_1) dV \quad (\text{B.4})$$

where ϕ_1 and ϕ_2 are arbitrary scalar functions and \mathbf{w}_1 and \mathbf{w}_2 are arbitrary vector functions.

Let ϕ_1 and \mathbf{w}_1 to be the solution of

$$\nabla \cdot \mathbf{w} = 0$$

$$\mathbf{w} = [B] \cdot \mathbf{u}'$$

$$\mathbf{u}' = \nabla \phi$$

and

$$\phi_2 = G$$

Doing so, Eq. (B.4) becomes

$$\iint_S (\phi[B] \nabla G - G[B] \nabla \phi) \cdot \mathbf{n} dS = \iiint_V \phi_1 \nabla[B] \nabla G dV$$

since $\nabla \cdot \mathbf{w}_2 = \nabla[B] \nabla \phi_2$. Applying Eq. (B.1), this becomes

$$\iint_S (\phi[B] \nabla G - G[B] \nabla \phi) \cdot \mathbf{n} dS = \iiint_V \phi_1 \delta(\mathbf{r} - \mathbf{r}_\infty) dV$$

Using finite-part integrals, this reduces to [21]

$$\phi(\mathbf{r}) = \iint_S \phi \frac{\partial G}{\partial \tilde{\mathbf{n}}} - G \frac{\partial \phi}{\partial \tilde{\mathbf{n}}} dS \quad (\text{B.5})$$

where

$$\frac{\partial \phi}{\partial \tilde{\mathbf{n}}} = \tilde{\mathbf{n}} \cdot \nabla \phi \quad (\text{B.6})$$

where $\tilde{\mathbf{n}}$ is the *conormal vector* to S , given by

$$\tilde{\mathbf{n}} = [B]\mathbf{n} \tag{B.7}$$

where \mathbf{n} is the normal vector to S pointing into V .

APPENDIX C

PILOT CODE

The code developed as part of this work is called MachLine (Multi-order Approach to Calculating High-speed Linear aerodynamics). It is freely available online via GitHub [118]. However, the code base will continue to be developed and expanded in the future as this work continues with other students and research objectives. Because of this, an archived copy of the code as it stood at the time of the publication of this work is available at [119].

C.1 Code Description

MachLine is written in Fortran incorporating elements up through the Fortran 2008 standard. It is object-oriented and contains user-defined types for handling such data objects as surface meshes, panels, wakes, and fluid flows. Most user-defined types within MachLine contain type-bound procedures which manipulate these types and perform calculations related to the type, such as calculating the potential induced by a panel at a given point in space.

The root directory for the source code is divided into five subdirectories:

- `src/` contains the Fortran files specific to MachLine.
- `common/` contains Fortran source files that are used in MachLine but are not necessarily specific to MachLine or even panel methods in general. Some of these files (`linked_list.f90`, `json.f90`, and `json_xtnsn.f90`) were taken from other open sources and modified as needed for MachLine.
- `test/` contains regression tests (implemented in Python) used during the development process.
- `dev/` contains miscellaneous files used during development.

- **studies/** contains the meshes and analysis scripts used to produce the sample results presented in this work, along with other studies.

The entry point for the code is **program main** defined in **src/main.f90**. This section of the code reads in the input file, instantiates the necessary surface mesh and solver types, initializes and runs the solver, and then directs any outputs based on the input settings. It primarily interacts with **body** (an instance of **surface_mesh**) and **linear_solver** (an instance of **panel_solver**). The solver is referred to as **linear_solver** because it was originally intended that MachLine would incorporate a transonic field-panel solver, for which a standard panel solver would be the “linear” part of the solution process.

The variable **body** stores all information about the user-provided mesh. The **surface_mesh** type is capable of loading mesh data from STL, VTK, and TRI files, automatically determining panel adjacency, automatically locating wake-shedding edges, placing interior control points, and other functions. As such, the user requirements for meshing are very light.

The variable **linear_solver** stores relatively little information. Its primary purpose is to bring together the data stored in other types to assemble and solve the linear system of equations and perform all post-processing. As such, it is one of the most involved user-defined types in MachLine.

The most involved type in MachLine is the **panel** type, which is defined in **src/panel.f90**. This type stores all necessary information for each panel (and potentially its mirror), such as location, area, strength-parameter transformations, global-to-local-scaled-coordinate transformations, and more. In addition, all calculations for determining the influence of a panel on a point are performed using procedures bound to the **panel** type.

C.2 Running the Code

MachLine is set up to be compiled using gfortran and Make. On a Linux machine with the necessary packages installed, simply executing `make` in the root directory of MachLine will compile the source code. Various compilation options exist for compiling the code in serial vs parallel, including certain debugging flags, etc.

Once the code has been compiled, it is run by executing `machline.exe <INPUT_FILENAME>`. If no input file is specified, the user will be prompted to input a file location.

Units and Coordinate Systems

MachLine is ignorant of any units of measure. It assumes that the user has given the inputs and will parse the results using a coherent unit system. MachLine also has no native coordinate system. It simply uses the coordinates of the mesh. Thus, in the output, the force coefficient `Cx` denotes the nondimensional force acting in the mesh x -direction, `CMx` denotes the nondimensional moment about the mesh x -axis, and so forth.

Input Files

The input files to MachLine are created using the JSON (JavaScript Object Notation) format. JSON files have a nested dictionary structure, where information is stored in key-value pairs. Each input file should contain the following five main sections:

- "flow" describes the freestream condition.
- "geometry" gives the location of the mesh file and some instructions on how it is to be handled.
- "solver" contains options for setting up, analyzing, and solving the linear system of equations, including which boundary condition is to be used.
- "post_processing" specifies primarily how surface pressures and loads should be calculated based on the solution to the linear system of equations.

- "output" provides MachLine with the desired file locations for storing the results of the analysis.

A basic MachLine input file looks like the following:

```
{
  "flow": {
    "freestream_velocity": [ 100.0, 0.0, -10.0 ]
  },
  "geometry": {
    "file": "my_swept_wing.vtk",
    "wake_model": {
      "wake_shedding_angle": 90.0,
      "trefftz_distance": 100.0,
      "N_panels": 1
    },
    "reference": {
      "area": 1.0
    }
  },
  "solver": {
    "formulation": "dirichlet-morino"
  },
  "post_processing" : {
    "pressure_for_forces": "prandtl-glauert",
    "pressure_rules": {
      "incompressible": true,
      "isentropic": false,
      "second-order": false
    }
  },
}
```

```

    "subsonic_pressure_correction": {
        "correction_mach_number": 0.8399,
        "prandtl-glauert": true,
        "karman-tsien": true,
        "laitone": true
    }
},
"output": {
    "body_file": "swept_wing_result.vtk",
    "wake_file": "wake_result.vtk",
    "control_point_file": "control_point_result.vtk",
    "report_file": "report.json"
}
}

```

MachLine has been designed to require minimal input, and so all the settings seen above are not required. For example, a full case can be run with the input file containing only the following information:

```

{
    "flow": {
        "freestream_velocity": [1.0, 0.0, 0.0]
    },
    "geometry": {
        "file": <my_swept_wing.vtk>
    },
    "solver": {},
    "post_processing" : {},
    "output": {
        "body_file": <swept_wing_result.vtk>
    }
}

```

```

    }
}

```

Of course, the user will often want to have more control over the execution of MachLine than this. However, this serves to demonstrate the lightweight user requirements of MachLine.

The settings available for each of the five sections of the input file will now be described. For nested settings (such as "wake_model" under "geometry"), if a parent is not specified, then all its children will adopt the default values. All settings which have no default must be specified.

Flow

The flow section describes the freestream flow into which the mesh is inserted. It is structured as follows:

```

{
    "flow" : {
        "freestream_velocity" : [<X VELOCITY>, <Y VELOCITY>, <Z VELOCITY>],
        "gamma" : <GAMMA>,
        "freestream_mach_number" : <MACH>
    },
    ...
}

```

The available options for each of these parameters are listed in Table C.1.

Geometry

The geometry section tells MachLine where the mesh is stored and how it should be treated. It is structured as follows:

```

{
    ...
    "geometry": {
        "file": <PATH TO MESH FILE>,

```

Table C.1: Options for the flow section of the input file.

Key	Value Type	Required?	Allowable Values	Default Value	Description
freestream- _velocity	list of floats	yes	arbitrary	N/A	Oncoming freestream velocity expressed in the mesh (global) coordinate system. Note the magnitude of the freestream velocity currently has no effect on the solution.
gamma	float	no	arbitrary	1.4	Freestream ratio of specific heats.
freestream- _mach_number	float	no	non- negative	0.0	Freestream Mach number. From this and the freestream velocity, MachLine will infer the freestream speed of sound and all other relevant flow variables.

```

"mirror_about" : <MIRROR PLANE>,
"singularity_order" : <"lower" OR "higher">,
"wake_model": {
    "wake_present" : <TRUE OR FALSE>,
    "append_wake" : <TRUE OR FALSE>,
    "wake_shedding_angle": <ANGLE>,
    "trefftz_distance": <DISTANCE>,
    "N_panels": <NUMBER OF WAKE PANELS>
},
"reference": {
    "area": <AREA>
}
},
...
}

```

The available options for each of these parameters are listed in Table C.2.

Solver

The solver section specifies how MachLine is to solve for the source and doublet strengths which uniquely specify the flow field. It is structured as follows:

```
{
...
  "solver": {
    "formulation" : <FORMULATION>,
    "control_point_offset" : <DISTANCE>,
    "control_point_offset_type" : <'direct' OR 'local'>,
    "run_checks" : <TRUE OR FALSE>,
    "matrix_solver" : <SOLVER>,
    "relaxation" : <RELAXATION FACTOR>,
    "tolerance" : <TERMINATION TOLERANCE>,
    "block_size" : <BLOCK SIZE>,
    "max_iterations" : <MAXIMUM SOLVER ITERATIONS>,
    "preconditioner" : <PRECONDITIONING METHOD>,
    "write_A_and_b" : <TRUE OR FALSE>,
    "sort_system" : <TRUE OR FALSE>,
    "iterative_solver_output" : <FILENAME>
  },
...
}
```

The available options for these parameters are listed in Table C.3.

Post-Processing

This is where pressure and force calculations are configured. It is structured as follows:

```

{
...
  "post_processing" : {
    "pressure_rules": {
      "incompressible": true,
      "isentropic": false,
      "second-order": false
    },
    "subsonic_pressure_correction": {
      "correction_mach_number": 0.8399,
      "prandtl-glauert": true,
      "karman-tsien": true,
      "laitone": true
    },
    "pressure_for_forces": "prandtl-glauert"
  },
...
}

```

The available options for these parameters are listed in Table C.4.

Output

The output section tells MachLine where to store the solver results. It is structured as follows:

```

{
...
  "output": {
    "verbose" : <TRUE OR FALSE>,
    "body_file" : <PATH TO FILE>,
    "mirrored_body_file" : <PATH TO FILE>,

```



```

    "wake_file" : <PATH TO FILE>,
    "control_point_file" : <PATH TO FILE>,
    "report_file" : <PATH TO FILE>,
    "offbody_points" : {
        "points_file" : <PATH TO FILE>,
        "output_file" : <PATH TO FILE>
    }
}
}

```

The available options for these parameters are listed in Table C.5.

For data visualization, the authors recommend ParaView, which can be downloaded from <https://www.paraview.org>.

C.3 Studies

Several studies into result obtained from MachLine for various configurations are included with with source code [119]. These are stored in the **studies/** directory. Each subdirectory of **studies/** contains one or more studies. For example, in the **sphere/** subdirectory, several studies regarding flow over a sphere are presented ranging from the effect of control point offset and mesh resolution on the flow results to a comparison against analytic solutions for compressible subsonic flow. Each study is a single Python script that will set up the necessary input files, run MachLine, and post-process the data from MachLine. The generated plots are then stored in the **plots/** subdirectory for viewing. All of the data presented in this dissertation were generated using these studies.

Since running many cases through MachLine can be expensive, within each study script is a global variable called **RERUN_MACHLINE**. If the user has previously run the study and only made changes to the post-processing, then this may be set to **False** to save the time it would take to reproduce all the data.

Most studies use the open-source ParaView Python modules for post-processing. Documentation for these modules, including installation instructions, may be found at <https://kitware.github.io/paraview-docs/latest/python/>.

Table C.2: Options for the geometry section of the input file.

Key	Value Type	Required?	Allowable Values	Default Value	Description
file	string	yes	arbitrary	N/A	Path to the mesh file relative to the current working directory.
mirror- _about	string	no	xy, xz, yz		Which plane to mirror the mesh about.
singularity- _order	string	no	lower, higher	lower	Sets the singularity distribution type for all panels.
max- _continuity- _angle	float	no	0.0-180.0	5.0	Maximum angle between panels before the edge is marked as discontinuous for higher-order distributions.
wake_model		no			
wake- _present	logical	no	true, false	true	Whether a wake should be assumed present. May be set to false for known nonlifting flows.
append- _wake	logical	no	true, false	value of wake- _present	Whether or not a wake should be explicitly added to the mesh.
wake- _shedding- _angle	float	no	0.0-180.0	90.0	Maximum allowable flow-turning angle between two panels before wake panels will be shed.
trefftz- _distance	float	no	non-negative	depends on mesh	Distance from the mesh origin at which the wake terminates.
N_panels	integer	no	non-negative	1	Half the number of panels distributed streamwise in the wake.
reference		no			
area	float	no	non-negative	1.0	Reference area.
length	float	no	non-negative	1.0	Reference length.
CG	vector	no	arbitrary	[0.0, 0.0, 0.0]	Center of gravity location described in mesh coordinates. The reported moment coefficients will be about this point.

Table C.3: Options for the solver section of the input file.

Key	Value Type	Required?	Allowable Values	Default Value	Description
formulation	string	no	dirichlet-morino, dirichlet-source-free	dirichlet-morino	Specifies the formulation which is to be used for enforcing boundary conditions.
control-point_offset	float	no	positive	10^{-7}	Distance at which control points are to be offset from the body surface.
control-point_offset_type	string	no	direct, local	direct	Method for offsetting control points from the mesh surface.
run_checks	logical	no	true, false	false	Whether to check the solvability of linear system.
matrix_solver	string	no	GMRES, QRUP, FQRUP, LU, BJAC, BSSOR	GMRES	Specifies which matrix solver is to be used for solving the linear system of equations.
relaxation	float	no	between 0 and 2	0.5	Relaxation factor used in the iterative matrix solvers.
tolerance	float	no	arbitrary	10^{-10}	Norm of the residual at which the iterative solvers terminate.
block_size	integer	no	arbitrary	100	Block size for the block iterative solvers.
max_iterations	integer	no	arbitrary	1000	Maximum number of iterations for the iterative solvers.
precondition	string	no	NONE, DIAG	DIAG	Preconditioning method to use on the linear system. DIAG specifies diagonal preconditioning.
write_A_and_b	logical	no	true, false	false	Whether to write the linear system to the files "A_mat.txt" and "b_vec.txt".
sort_system	logical	no	true, false	true	Specifies whether the linear system of equations should be sorted; this is necessary for the QRUP and FQRUP solvers.

Table C.4: Options for the post-processing section of the input file.

Key	Value Type	Required?	Allowable Values	Default Value	Description
pressure- _rules		no			
incompressible	logical	no	true, false	true if $M = 0$, false otherwise	Whether to use the incompressible pressure rule.
isentropic	logical	no	true, false	true if $M > 0$, false otherwise	Whether to use the isentropic pressure rule.
second-order	logical	no	true, false	false	Whether to use the second-order pressure rule.
slender-body	logical	no	true, false	false	Whether to use the slender-body pressure rule.
linear	logical	no	true, false	false	Whether to use the linear pressure rule.
subsonic- _pressure- _corrections		no			
correction- _mach- _number	float	no	non-negative	0.0	Freestream Mach number used to correct incompressible pressure coefficients for compressibility.
prandtl- glauert	logical	no	true, false	false	Whether to use the Prandtl-Glauert rule to correct pressure coefficients for compressibility.
karman-tsien	logical	no	true, false	false	Whether to use the Karman-Tsien rule to correct pressure coefficients for compressibility.
laitone	logical	no	true, false	false	Whether to use the Laitone rule to correct pressure coefficients for compressibility.
pressure-_for- _forces	string	no	any pressure rule or correction	highest-order available	Specifies which pressure is to be used for total force integration.

Table C.5: Options for the output section of the input file.

Key	Value Type	Required?	Allowable Values	Default Value	Description
verbose	logical	no	true, false	false	Whether solver messages should be written to the terminal during execution.
body_file	string	no	arbitrary	none	File location where the surface mesh results should be stored.
mirrored-body_file	string	no	arbitrary	none	File location where the mirrored surface mesh results should be stored.
wake_file	string	no	arbitrary	none	File location where the wake mesh results should be stored.
control-point_file	string	no	arbitrary	none	File location where the control point results should be stored.
mirrored-control-point_file	string	no	arbitrary	none	File location where the mirrored control point results should be stored.
report_file	string	no	arbitrary	none	File location where a report on the output of the solver will be stored.
offbody-points		no			
points_file	string	no	arbitrary	none	File location containing a list of points at which MachLine is to calculate velocity potentials and velocities. Must be formatted as comma-separated values. Each line is one point. MachLine assumes this file has a one-line header.
output_file	string	no	arbitrary	none	File location to write velocity potentials and velocities to. These will be calculated at the points specified in points_file.

APPENDIX D

PSEUDOCODE FOR THE DOMAIN OF DEPENDENCE ALGORITHMS

The pseudocode for the brute-force domain of dependence algorithm is given in Algorithm 8. Note the extreme simplicity of this algorithm and the fact that it may be easily parallelized.

Algorithm 8 Brute-force method for determining point domains of dependence.

```

1: Input
2:    $v$    Vector of  $N$  mesh vertices. Each vertex stores its location ( $loc$ ) and the list of
        all other vertices within its domain of dependence ( $D$ )
3:  $i \leftarrow 0$ 
4: while  $i < N$  do
5:    $i \leftarrow i + 1$ 
6:    $j \leftarrow 0$ 
7:   while  $j < N$  do
8:      $j \leftarrow j + 1$ 
9:     if  $loc_j$  is in the DoD for  $v_i$  (based on Eqs. (4.1) and (4.2)) then
10:      Append  $j$  to  $D_i$ 
11:     end if
12:   end while
13: end while

```

The pseudocode for the recursive domain of dependence algorithm is given in Algorithm 9. This algorithm was implemented in Python, which allowed for using the set functions (union and intersection) directly. Were this algorithm to be implemented in a different language, it might need to be altered to account for differing available storage structures.

As described in the body of this work, the recursive algorithm requires keeping track of all upstream vertices not yet known to be within a given vertex's domain of dependence. This is done here using E for each vertex. Initially, E_i is set to contain all vertices upstream of vertex i (see line 7 in Algorithm 9). Then, when a given vertex is found to belong to the domain of dependence for vertex i , it is removed from E_i . Once D has been determined for

Algorithm 9 Recursive method for determining point domains of dependence.

```

1: Input
2:    $v$    Vector of  $N$  mesh vertices. Each vertex stores its location ( $loc$ ), the list of all
        other vertices within its domain of dependence ( $D$ ), and the list of all vertices not within
        its domain of dependence  $E$ 
3: Sort  $v$  based on Eq. (4.4)
4:  $i \leftarrow 0$ 
5: while  $i < N$  do
6:    $i \leftarrow i + 1$ 
7:    $E_i \leftarrow [1 \dots i - 1]$  ▷ Initialize  $E$  with all upstream vertices
8:    $j \leftarrow 1$ 
9:   while  $j < \text{length of } E_i$  do
10:     $k \leftarrow E_{ij}$  ▷ Get the  $k$ -th vertex from  $E$ 
11:    if  $loc_k$  is in the DoD for  $v_i$  (based on Eqs. (4.1) and (4.2)) then
12:       $D_i \leftarrow D_i \cup D_k$  ▷ All vertices in  $D_k$  are now known to belong to  $D_i$  as well
13:      Remove  $k$  from  $E_i$  and append  $k$  to  $D_i$ 
14:       $E_i \leftarrow E_i \cap E_k$  ▷ Remove known vertices from  $E_i$ 
15:    else
16:       $j \leftarrow j + 1$  ▷ If a vertex wasn't moved over, then go to the next vertex
17:    end if
18:  end while
19: end while
20:  $i \leftarrow 0$ 
21: while  $i < N$  do
22:    $i \leftarrow i + 1$ 
23:    $E_i \leftarrow E_i \cap [i + 1 \dots N]$  ▷ Once everything else is known, add all downstream
        vertices to  $E$ 
24: end while

```

each vertex, all vertices downstream of each vertex are added to E (see line 24 in Algorithm 9). These vertices are known to be outside the domain of dependence, and so did not need to be considered in the recursive algorithm.

APPENDIX E

INFLUENCE COEFFICIENT CALCULATION DEVELOPMENT METHODOLOGY

In implementing the influence coefficient calculation procedure used in MachLine, a number of different approaches were taken. This was due mainly to the fact that (as described previously) various methods for calculating supersonic panel influences are described in the relevant literature. To review briefly, there is the Epton-Magnus method, described in the PAN AIR documentation [33], which unifies influence calculations for supersonic and subsonic flows [33]. There is also the Ehlers-Davis method, described by Ehlers et al. [2] and implemented by Davis in CPanel [7], which applies only to supersonic flow over subinclined panels. Finally, there is the Ehlers-Johnson method, described in Appendix E of [2] and implemented in the PAN AIR source code for supersonic, subinclined panels, which uses the nomenclature first introduced by Johnson [45]. Ehlers et al. also presented a method for superinclined panels [2]; no implementations of this method implemented have been found.

Initially, MachLine was developed as a subsonic-only code. For this, the influence coefficient calculation method described by Johnson [45] was used. Johnson's method for subsonic flows is well-documented, and so the implementation of this method went smoothly.

To add supersonic capabilities to MachLine, the Epton-Magnus method was first implemented, as it seemed to be the most general and well-documented of all the methods available. The implementation was debugged based on pressure calculations over a rectangular wing with a diamond section, the qualitative behavior of which was well-understood. Unfortunately, significant inconsistency and error were quickly discovered in the Epton-Magnus method as described in [33], and the results coming from MachLine with this method implemented were in error.

Upon examining the PAN AIR source code, it was discovered that the Ehlers-Johnson method was used for the calculation of influence coefficients for subinclined panels in supersonic flow. Thus, this method was next implemented in MachLine. To implement this

method, both the description given by Ehlers et al. [2] and the PAN AIR source code [65] were used. Unfortunately, it was discovered that the description given by Ehlers et al. was inconsistent with what was implemented in PAN AIR. In addition, the method as implemented produced incorrect results.

At this point, it was decided to attempt the Ehlers-Davis method, since this method had successfully been implemented in recent years as part of an unstructured panel method [7]. While ultimately unsuccessful, several important principles were learned from attempting to implement this method. One was that this method, as thoroughly explained by Davis [7], gives different influence formulas based on whether both, one, or neither of an edge's endpoints falls inside the domain of dependence. This is due to that fact that the influence coefficient formulas go to zero on the Mach cone [7]. This fact needed to be understood in order to develop a viable influence calculation procedure. This made clear the reasoning behind some of the calculations performed in the PAN AIR source code that were not explicitly described by Ehlers et al. [2].

Another lesson learned from implementing the Ehlers-Davis method was the need for testing the influence calculation procedure at a smaller scale before integrating it with the larger panel method. As with the Ehlers-Johnson method, the Ehlers-Davis method was first tested based on the overall results obtained on a diamond wing. However, it was difficult to verify the implementation of the formulas this way, and so, following the example of Davis, a different approach was taken. The implementation was shifted to Python, and the focus shifted to the influence of a single panel on a single point in space. This allowed comparing the influence of a panel to the influence of a point singularity, which should be approximately equal far from the panel. Nearer to the panel, these influences should be qualitatively similar.

Unfortunately, even with switching to a single panel and single control point, the Ehlers-Davis method could not be implemented such that it provided satisfactory influence calculations. This was mainly due to ambiguity in the calculation of the oblique, edge-based coordinate system. The formulas given by Ehlers et al. were found to be internally

inconsistent [2], and the proper form for these formulas could not be determined based on the purely qualitative descriptions given by Davis [7].

Taking the lessons learned from the Ehlers-Davis method, the Ehlers-Johnson method was tried again. Upon implementing this method, accurate results compared to the point singularities (both source and doublet) were obtained for the case of the panel falling entirely inside the domain of dependence. This was the case for both panels with purely supersonic edges and with purely subsonic edges. In addition to the lessons learned above, success was found by recognizing the difference in convention for the direction of doublet-induced flow between panel methods and potential flow in general (discussed in Section 1.3.3) and the flow-dependence of the sign of the doublet influence (Eq. (5.12)).

Debugging the implementation for the formulas for panels partially intersecting the domain of dependence was more difficult. In this case, the panel and point influence calculations do not match qualitatively over all possible control point locations due to the fact that the calculation of panel influence coefficients relies on the use of Hadamard finite part integrals [21]. Because of this, the influence coefficients calculated for a panel remain finite even at points where the panel intersects the domain of dependence. On the other hand, the influence of a point singularity is clearly singular when the control point is on the downstream Mach cone. In verifying these formulas, the PAN AIR source code was particularly valuable.

Following this method for the superinclined panel influence coefficient calculations made development progress much more quickly. For superinclined panels, the influence calculation procedure described by Epton and Magnus [33] and used in the PAN AIR source code was duplicated.

The above method worked well for verifying the influence coefficients for constant source and doublet panels. In these cases, only $F(1, 1, 1)$, $hH(1, 1, 3)$, and $H(1, 1, 1)$ needed to be calculated. However, the influence of higher-order distributions could not be approximated by a single point singularity. When verifying the proper formulas for higher-order source and doublet distributions, it was useful to compare the panel influence calculations to

discrete distributions of point sources and doublets. Again using Python, classes of panels containing a variable number of point sources and doublets were created. Since multiple point singularities were used, the effect of doublet and source strength varying across the panel surface could be accounted for. Comparing these results with the analytic formulas allowed for rapid debugging of the higher-order influence integrals.

In addition to using distributions of discrete sources and doublets for comparing the influence integrals, Gaussian quadrature was used to verify the final formulas. Gaussian quadrature was useful because it provided a result that was exactly equal to the result from the analytic formulas if the analytic formulas were implemented correctly. However, Gaussian quadrature could not be used to evaluate the singular integrals that would appear if the panel partially intersected the domain of dependence. For such cases, the distributions of discrete sources and doublets provided a satisfactory method for qualitative verification.

Given the difficulty of the procedure just described, it is worth asking why the formulas for $hH(1, 1, 3)$ and $F(1, 1, 1)$ were not just derived from scratch. This would have been possible but would have required knowledge of fairly complex mathematics, particularly finite-part integrals. Even the original developers of these methods stated that certain formulas were only obtained “upon considerable analysis” [2]. It was deemed more efficient to implement and debug the influence calculation procedures as described in the existing literature than to derive them from scratch.

APPENDIX F

DERIVATION OF A USEFUL INTEGRAL IDENTITY

An integral identity is derived here which is often used in the literature regarding panel methods. This identity is typically applied when calculating panel influence coefficients (for example, see p. 23 in [50]). This identity is taken from a form of Stoke's theorem, expressed here as [149]

$$\iint_R \frac{\partial V}{\partial x} dx dy = \int_C V(x, y) dy \quad (\text{F.1})$$

Let V be the product of two functions (i.e. $V = UW$). Then, it may be written (using the product rule)

$$\iint_R \frac{\partial}{\partial x}(UW) dx dy = \iint_R U \frac{\partial W}{\partial x} dx dy + \iint_R W \frac{\partial U}{\partial x} dx dy = \int_C UW dy \quad (\text{F.2})$$

This may be rearranged to yield

$$\iint_R U \frac{\partial W}{\partial x} dx dy = \int_C UW dy - \iint_R W \frac{\partial U}{\partial x} dx dy \quad (\text{F.3})$$

Recognizing that $dy = \hat{t}_y dl$, where \hat{t}_y is the y -component of the unit tangent vector to the curve C , this becomes

$$\iint_R U \frac{\partial W}{\partial x} dx dy = \int_C UW \hat{t}_y dl - \iint_R W \frac{\partial U}{\partial x} dx dy \quad (\text{F.4})$$

The integration is now performed in terms of distance along C , rather than y .

For a derivative taken in the y direction, the required analog of Eq. (F.1) is

$$\iint_R \frac{\partial V}{\partial y} dx dy = \int_C -V(x, y) dx \quad (\text{F.5})$$

and the integral identity becomes

$$\iint_R U \frac{\partial W}{\partial y} dx dy = - \int_C UW \hat{t}_x dl - \iint_R W \frac{\partial U}{\partial y} dx dy \quad (\text{F.6})$$

Notice the change in sign due to the use of the cross product in Stoke's theorem.

As a consequence of these identities, the panel influence coefficient calculations (Chapter 5) are often presented in terms of sums of line and surface integrals (see for example [2, 33, 45, 50]).

APPENDIX G

PROOF OF SUBSONIC $F(1, 1, 1)$ FORMULA

Previously, it was claimed that

$$F(1, 1, 1) = \begin{cases} \ln \left(\frac{(\sqrt{l_1^2 + g^2} - l_1)(\sqrt{l_2^2 + g^2} + l_2)}{g^2} \right), & \text{sign}(l_1) \neq \text{sign}(l_2) \\ \text{sign}(l_1) \ln \left(\frac{\sqrt{l_2^2 + g^2} + |l_2|}{\sqrt{l_1^2 + g^2} + |l_1|} \right), & \text{otherwise} \end{cases} \quad (\text{G.1})$$

is equivalent to the formulas for $F(1, 1, 1)$ in subsonic flow given by Johnson, namely [45]

$$F(1, 1, 1) = \begin{cases} \ln \left(\frac{(\sqrt{l_1^2 + g^2} - l_1)(\sqrt{l_2^2 + g^2} + l_2)}{g^2} \right), & \text{sign}(l_1) \neq \text{sign}(l_2) \\ \ln \left(\frac{\sqrt{l_2^2 + g^2} + l_2}{\sqrt{l_1^2 + g^2} + l_1} \right), & l_1, l_2 \geq 0 \\ \ln \left(\frac{\sqrt{l_1^2 + g^2} - l_1}{\sqrt{l_2^2 + g^2} - l_2} \right), & l_1, l_2 < 0 \end{cases} \quad (\text{G.2})$$

The first case is clearly identical to that given by Johnson. Consider now the case when $l_1, l_2 < 0$. Since both are negative, it must be that

$$F(1, 1, 1) = \ln \left(\frac{\sqrt{l_1^2 + g^2} - l_1}{\sqrt{l_2^2 + g^2} - l_2} \right) = \ln \left(\frac{\sqrt{l_1^2 + g^2} + |l_1|}{\sqrt{l_2^2 + g^2} + |l_2|} \right) \quad (\text{G.3})$$

Using the properties of the logarithm of a quotient, this becomes

$$F(1, 1, 1) = -\ln \left(\frac{\sqrt{l_2^2 + g^2} + |l_2|}{\sqrt{l_1^2 + g^2} + |l_1|} \right) = \text{sign}(l_1) \ln \left(\frac{\sqrt{l_2^2 + g^2} + |l_2|}{\sqrt{l_1^2 + g^2} + |l_1|} \right) \quad (\text{G.4})$$

since $l_1 < 0$.

When $l_1, l_2 \geq 0$, it is trivially true that

$$F(1, 1, 1) = \text{sign}(l_1) \ln \left(\frac{\sqrt{l_2^2 + g^2} + |l_2|}{\sqrt{l_1^2 + g^2} + |l_1|} \right) = \ln \left(\frac{\sqrt{l_2^2 + g^2} + l_2}{\sqrt{l_1^2 + g^2} + l_1} \right) \quad (\text{G.5})$$

Thus, the formula implemented in MachLine is equivalent to that given by Johnson [45]. Multiplying the entire logarithm by $\text{sign}(l_1)$ only works as intended if $\text{sign}(0) = 1$ (otherwise, $l_1 = 0$ would mean $F(1, 1, 1) = 0$). This is the case with Fortran, the language used to write MachLine.

APPENDIX H

H AND F INTEGRAL RECURSION RELATIONS

In Chapter 5, it was shown how the necessary integrals for determining the influence of a source and doublet panel may be calculated based on the values of two fundamental integrals, $hH(1, 1, 3)$ and $F(1, 1, 1)$. The integral relations used there were given without proof. Here, it will be shown how these basic relationships may be derived. Throughout this section, the local-scaled system is assumed to simplify notation. In addition, whenever an F integral appears in a relation along with H integrals, the summation of the F integral over all edges is implied.

H.1 Definitions

First, recall

$$H(M, N, K) = \iint_S \frac{(\xi - x)^{M-1} (\eta - y)^{N-1}}{R^K} d\xi d\eta \quad (\text{H.1})$$

where

$$R = \sqrt{r(\xi - x)^2 + s(\eta - y)^2 + rsh^2} \quad (\text{H.2})$$

and

$$F_i(M, N, K) = \int_{\text{edge } i} \frac{(\xi - x)^{M-1} (\eta - y)^{N-1}}{R^K} dl_i \quad (\text{H.3})$$

where $\mathbf{Q} = (\xi, \eta, 0)$ is the point of integration on the panel, $\mathbf{P} = (x, y, z)$ is the evaluation point, r is the panel inclination indicator, s is the flow type indicator, and

$$h = z \quad (\text{H.4})$$

Also recall the edge outward normal vector

$$\hat{\mathbf{n}} = \begin{Bmatrix} \hat{t}_\eta \\ -\hat{t}_\xi \end{Bmatrix} \quad (\text{H.5})$$

where $\hat{\mathbf{t}}$ is the edge tangent vector. From this, the in-plane edge perpendicular distance, a , is defined as

$$a = \hat{n}_\xi(\xi - x) + \hat{n}_\eta(\eta - y) \quad (\text{H.6})$$

H.2 H Recursion Relations

The second recursion relation relates only H integrals:

$$H(M, N, K - 2) = \iint_S \frac{(\xi - x)^{M-1} (\eta - y)^{N-1}}{R^K} R^2 d\xi d\eta$$

$$H(M, N, K - 2) = \iint_S \frac{(\xi - x)^{M-1} (\eta - y)^{N-1}}{R^K} \left[r (\xi - x)^2 + s (\eta - y)^2 + rsh^2 \right] d\xi d\eta$$

$$H(M, N, K - 2) = rH(M + 2, N, K) + sH(M, N + 2, K) + rsh^2 H(M, N, K) \quad (\text{H.7})$$

Now using the integral identity established in Appendix F, two more recursion relations may be derived. First, it may be easily shown that

$$\frac{\partial}{\partial \xi} \left(\frac{(\xi - x)^{M-1} (\eta - y)^{N-1}}{R^K} \right) = (M - 1) \frac{(\xi - x)^{M-2} (\eta - y)^{N-1}}{R^K} - rK \frac{(\xi - x)^M (\eta - y)^{N-1}}{R^{K+2}}$$

Then, setting $U = 1$ and $W = \frac{(\xi - x)^{M-1} (\eta - y)^{N-1}}{R^K}$ in Eq. (F.4) yields

$$\begin{aligned} & \iint_S (M-1) \frac{(\xi-x)^{M-2} (\eta-y)^{N-1}}{R^K} - rK \frac{(\xi-x)^M (\eta-y)^{N-1}}{R^{K+2}} d\xi d\eta \\ &= \sum_{\text{edges}} \int \frac{(\xi-x)^{M-1} (\eta-y)^{N-1}}{R^K} \hat{t}_\eta dl \end{aligned}$$

From the definition of $\hat{\mathbf{n}}$ and the H and F integrals, and shifting the M and K indices for convenience, this may then be written as

$$(M-2)H(M-2, N, K-2) - r(K-2)H(M, N, K) = \hat{n}_\xi F(M-1, N, K-2) \quad (\text{H.8})$$

Using a similar procedure in applying Eq. (F.6) and recognizing

$$\frac{\partial}{\partial \eta} \left(\frac{(\xi-x)^{M-1} (\eta-y)^{N-1}}{R^K} \right) = (N-1) \frac{(\xi-x)^{M-1} (\eta-y)^{N-2}}{R^K} - sK \frac{(\xi-x)^{M-1} (\eta-y)^N}{R^{K+2}}$$

the relation

$$(N-2)H(M, N-2, K-2) - s(K-2)H(M, N, K) = \hat{n}_\eta F(M, N-1, K-2) \quad (\text{H.9})$$

may be derived.

Now let Eq. (H.8) be substituted into Eq. (H.7) for $H(M+2, N, K)$ and Eq. (H.9) for $H(M, N+2, K)$. Doing so yields

$$\begin{aligned} H(M, N, K-2) &= \frac{1}{K-2} [-\hat{n}_\xi F(M+1, N, K-2) + MH(M, N, K-2)] \\ &\quad + \frac{1}{K-2} [-\hat{n}_\eta F(M, N+1, K-2) + NH(M, N, K-2)] \\ &\quad + rsh^2 H(M, N, K) \end{aligned}$$

Gathering like terms and using Eq. (H.11), this may be written

$$(K - M - N - 2)H(M, N, K - 2) = -aF(M, N, K - 2) + (K - 2)rs h^2 H(M, N, K) \quad (\text{H.10})$$

By means of Eqs. (H.7), (H.8), (H.9), and (H.10), all integrals necessary for determining panel influence coefficients may be determined.

H.3 F Recursion Relations

Multiplying the definition of a by the integrand in Eq. (H.3) yields

$$a \frac{(\xi - x)^{M-1}(\eta - y)^{N-1}}{R^K} = \hat{n}_\xi(\xi - x) \frac{(\xi - x)^{M-1}(\eta - y)^{N-1}}{R^K} + \hat{n}_\eta(\eta - y) \frac{(\xi - x)^{M-1}(\eta - y)^{N-1}}{R^K}$$

Taking the integral of both sides over the panel edge and recognizing that a is constant with respect to the integration, this becomes

$$aF(M, N, K) = \hat{n}_\xi F(M + 1, N, K) + \hat{n}_\eta F(M, N + 1, K) \quad (\text{H.11})$$

Another F recursion relation may be obtained by considering the quantity [45]

$$E(\xi, \eta) = \frac{(\xi - x)^{M-1}(\eta - y)^{N-1}}{R^{K-2}} \quad (\text{H.12})$$

on a panel edge. The derivative of this quantity with respect to the displacement along the edge may be found using the chain rule, resulting in

$$\frac{\partial E}{\partial l} = \frac{\partial E}{\partial \xi} \frac{\partial \xi}{\partial l} + \frac{\partial E}{\partial \eta} \frac{\partial \eta}{\partial l} \quad (\text{H.13})$$

Taking the line integral of this quantity along the length of the edge and using the definition of $\hat{\mathbf{n}}$ yields

$$\int \frac{\partial E}{\partial l} dl = -\hat{n}_\eta \int \frac{\partial E}{\partial \xi} dl + \hat{n}_\xi \int \frac{\partial E}{\partial \eta} dl \quad (\text{H.14})$$

From the definition of E , it may be found

$$\frac{\partial E}{\partial \xi} = (M-1) \frac{(\xi-x)^{M-2}(\eta-y)^{N-1}}{R^{K-2}} - r(K-2) \frac{(\xi-x)^M(\eta-y)^{N-1}}{R^K}$$

$$\frac{\partial E}{\partial \eta} = (N-1) \frac{(\xi-x)^{M-1}(\eta-y)^{N-2}}{R^{K-2}} - s(K-2) \frac{(\xi-x)^{M-1}(\eta-y)^N}{R^K}$$

Using these in Eq. (H.14) yields

$$\begin{aligned} E(M, N, K-2) = & -\hat{n}_\eta [(M-1)F(M-1, N, K-2) - r(K-2)F(M+1, N, K)] \\ & + \hat{n}_\xi [(N-1)F(M, N-1, K-2) - s(K-2)F(M, K+1, K)] \end{aligned} \quad (\text{H.15})$$

where

$$E(M, N, K) = \left. \frac{(\xi-x)^{M-1}(\eta-y)^{N-1}}{R^K} \right|_1^2 \quad (\text{H.16})$$

Now solving Eq. (H.11) for $F(M+1, N, K)$ yields

$$F(M+1, N, K) = (aF(M, N, K) - \hat{n}_\eta F(M, N+1, K)) / \hat{n}_\xi$$

Using this in Eq. (H.15) then results in

$$\begin{aligned} E(M, N, K-2) = & -\hat{n}_\eta [(M-1)F(M-1, N, K-2) \\ & - r(K-2) (aF(M, N, K) - \hat{n}_\eta F(M, N+1, K)) / \hat{n}_\xi] \\ & + \hat{n}_\xi [(N-1)F(M, N-1, K-2) - s(K-2)F(M, N+1, K)] \end{aligned}$$

Multiplying both sides by \hat{n}_ξ will then give

$$\begin{aligned}
\hat{n}_\xi E(M, N, K-2) = & -\hat{n}_\eta \hat{n}_\xi (M-1)F(M-1, N, K-2) \\
& + r\hat{n}_\eta a(K-2)F(M, N, K) - r\hat{n}_\eta^2(K-2)F(M, N+1, K) \\
& + \hat{n}_\xi^2(N-1)F(M, N-1, K-2) - s\hat{n}_\xi^2(K-2)F(M, N+1, K)
\end{aligned}$$

$$\begin{aligned}
\hat{n}_\xi E(M, N, K-2) = & -\hat{n}_\eta \hat{n}_\xi (M-1)F(M-1, N, K-2) + r\hat{n}_\eta a(K-2)F(M, N, K) \\
& + \hat{n}_\xi^2(N-1)F(M, N-1, K-2) - (s\hat{n}_\xi^2 + r\hat{n}_\eta^2)(K-2)F(M, N+1, K)
\end{aligned}$$

Rearranging, this becomes

$$\begin{aligned}
F(M, N+1, K) = & \frac{1}{(s\hat{n}_\xi^2 + r\hat{n}_\eta^2)(K-2)} [-\hat{n}_\eta \hat{n}_\xi (M-1)F(M-1, N, K-2) \\
& + r\hat{n}_\eta a(K-2)F(M, N, K) \\
& + \hat{n}_\xi^2(N-1)F(M, N-1, K-2) \\
& - \hat{n}_\xi E(M, N, K-2)]
\end{aligned} \tag{H.17}$$

Following a similar procedure for $F(M+1, N, K)$ results in

$$F(M, N+1, K) = (aF(M, N, K) - \hat{n}_\xi F(M+1, N, K)) / \hat{n}_\eta$$

$$\begin{aligned}
E(M, N, K-2) = & -\hat{n}_\eta [(M-1)F(M-1, N, K-2) - r(K-2)F(M+1, N, K)] \\
& + \hat{n}_\xi [(N-1)F(M, N-1, K-2) \\
& - s(K-2)(aF(M, N, K) - \hat{n}_\xi F(M+1, N, K)) / \hat{n}_\eta]
\end{aligned}$$

$$\begin{aligned}
\hat{n}_\eta E(M, N, K-2) = & -\hat{n}_\eta^2(M-1)F(M-1, N, K-2) + r\hat{n}_\eta^2(K-2)F(M+1, N, K) \\
& + \hat{n}_\xi \hat{n}_\eta(N-1)F(M, N-1, K-2) - s\hat{n}_\xi(K-2)aF(M, N, K) \\
& + s\hat{n}_\xi^2(K-2)F(M+1, N, K)
\end{aligned}$$

$$\begin{aligned}
\hat{n}_\eta E(M, N, K-2) = & -\hat{n}_\eta^2(M-1)F(M-1, N, K-2) + \hat{n}_\xi \hat{n}_\eta(N-1)F(M, N-1, K-2) \\
& - s\hat{n}_\xi(K-2)aF(M, N, K) + (s\hat{n}_\xi^2 + r\hat{n}_\eta^2)(K-2)F(M+1, N, K)
\end{aligned}$$

$$\begin{aligned}
F(M+1, N, K) = & \frac{1}{(s\hat{n}_\xi^2 + r\hat{n}_\eta^2)(K-2)} [-\hat{n}_\xi \hat{n}_\eta(N-1)F(M, N-1, K-2) \\
& + s\hat{n}_\xi a(K-2)F(M, N, K) \\
& + \hat{n}_\eta^2(M-1)F(M-1, N, K-2) \\
& + \hat{n}_\eta E(M, N, K-2)]
\end{aligned} \tag{H.18}$$

These last two relations may then be used to calculate $F(2, 1, 1)$ and $F(1, 2, 1)$ from $F(1, 1, 1)$ and $E(1, 1, -1)$, which may be calculated directly. The exception to this is the case of nearly-sonic edges on subinclined panels in supersonic flow. For these, the quantity $s\hat{n}_\xi^2 + r\hat{n}_\eta^2$ is close to zero, making these formulas unstable. For these calculations, a series expansion is instead used, as described by Ehlers et al. [2].

APPENDIX I

CALCULATION OF C INTEGRALS

Integrating a quadratic pressure distribution over a flat panel requires the calculation of a set of integrals of the form

$$C_{ij} = \iint_{S_i} \xi^i \eta^j d\xi d\eta \quad (\text{I.1})$$

where (ξ, η) is a point on the panel described in local-scaled coordinates. Applying Gauss' theorem to Eq. (I.1) (same as (7.26)) results in [33]

$$\iint_{S_i} \xi^i \eta^j d\xi d\eta = \frac{1}{i+1} \iint_{S_i} \frac{\partial}{\partial \xi} (\xi^{i+1} \eta^j) d\xi d\eta = \frac{1}{i+1} \int_{\partial S_i} \hat{n}_{x^{ls}} \xi^{i+1} \eta^j ds \quad (\text{I.2})$$

where ∂S_i is the boundary of the panel surface S_i . Breaking this integral up over the three edges of the panel results in [33]

$$C_{ij} = \frac{1}{i+1} \sum_{k=1}^3 \int_{\text{edge}_k} \hat{n}_{x^{ls}} \xi^{i+1} \eta^j ds_k \quad (\text{I.3})$$

Making a change of variables in the integration, this may be simplified to [33]

$$C_{ij} = \frac{1}{i+1} \sum_{k=1}^3 \Delta \eta_k \int_0^1 \xi^{i+1} \eta^j d\tau \quad (\text{I.4})$$

where τ is the position along the edge nondimensionalized by the edge length and $\Delta \eta_k$ is the difference in y^{ls} coordinates for the endpoints of edge k .

Let

$$G_{ij}^k = \int_0^1 \xi^i \eta^j d\tau \quad (\text{I.5})$$

for edge k . With this definition, C_{ij} may be found from [33]

$$C_{ij} = \sum_{k=1}^3 \frac{\Delta\eta_k}{i+1} G_{i+1,j}^k \quad (\text{I.6})$$

If the edge begins at (ξ_0, η_0) , then

$$\xi = \xi_0 + \Delta\xi\tau \quad (\text{I.7})$$

$$\eta = \eta_0 + \Delta\eta\tau \quad (\text{I.8})$$

on the edge.

Dropping the k superscript, let

$$H_{ij} = \int_0^1 \tau^i \eta^j d\tau \quad (\text{I.9})$$

for an edge. It will first be shown how these H integrals may be calculated, and then it will be shown how G_{ij} may be calculated from H_{ij} by means of another intermediate integral.

Evaluating the H integrals for $j = 0$ is simple, and results in

$$H_{i,0} = \frac{1}{i+1} \quad (\text{I.10})$$

Then, using Eq. (I.8), H_{ij} may be written as

$$H_{ij} = \int_0^1 \tau^i \eta^{j-1} (\eta_0 + \Delta\eta\tau) d\tau$$

$$H_{ij} = \eta_0 \int_0^1 \tau^i \eta^{j-1} d\tau + \Delta\eta \int_0^1 \tau^{i+1} \eta^{j-1} d\tau$$

$$H_{ij} = \eta_0 H_{i,j-1} + \Delta\eta H_{i+1,j-1} \quad (\text{I.11})$$

All necessary H integrals may be calculated by first calculating H_{i0} for all needed i values and then using Eq. (I.11) to calculate the rest of the H_{ij} integrals increasing in j .

Now let

$$I_{ijk} = \int_0^1 \xi^k \tau^{i-k} \eta^j d\tau \quad (\text{I.12})$$

Immediately, it may be seen that

$$G_{ij} = I_{iji} \quad (\text{I.13})$$

$$H_{ij} = I_{ij0} \quad (\text{I.14})$$

Using Eq. (I.7), I may be written as

$$\begin{aligned} I_{ijk} &= \int_0^1 \xi^{k-1} (\xi_0 + \Delta\xi\tau) \tau^{i-k} \eta^j d\tau \\ I_{ijk} &= \xi_0 \int_0^1 \xi^{k-1} \tau^{i-k} \eta^j d\tau + \Delta\xi \int_0^1 \xi^{k-1} \tau^{i-k+1} \eta^j d\tau \\ I_{ijk} &= \xi_0 I_{i-1,j,k-1} + \Delta\xi I_{i,j,k-1} \end{aligned} \quad (\text{I.15})$$

Once the necessary H integrals have been calculated, Eq. (I.15) may be used recursively to calculate the necessary G integrals.

In order to integrate the quadratic pressure distribution over a panel, C_{ij} is needed for $i = 0, 1, 2$ and $j = 0, 1, 2$ (except C_{12} , C_{21} , and C_{21} are not needed). From Eq. (I.6), this means G_{ij} is needed for $i = 1, 2, 3$, and $j = 0, 1, 2$ for each edge. This requires calculating I_{ijk} for $i = 1, 2, 3, 4$, $j = 0, 1, 2$, and $k = 0, 1, 2, 3$, and H_{ij} for $i = 1, 2, 3$ and $j = 0, 1, 2$. The following algorithm (a variation of that presented in [33]) is provided for calculating these values.

Algorithm 10 Method for calculating C_{ij} for a triangular panel.

```

1: Input
2:    $\xi_0^k$  Start point in  $\xi$  for edge  $k$ 
3:    $\Delta\xi^k$  Displacement in  $\xi$  for edge  $k$ 
4:    $\eta_0^k$  Start point in  $\eta$  for edge  $k$ 
5:    $\Delta\eta^k$  Displacement in  $\eta$  for edge  $k$ 
6: Output
7:    $C_{ij}$  Matrix of  $C$  integrals necessary for integrating the quadratic pressure distribution
8: while  $k \leq 3$  do                                     ▷ Loop through edges
9:    $i \leftarrow 1$                                            ▷ Initialize  $H_{i0}^k$ 
10:  while  $i \leq 4$  do
11:     $H_{i0}^k \leftarrow \frac{1}{i+1}$ 
12:     $i \leftarrow i + 1$ 
13:  end while
14:   $i \leftarrow 1$                                            ▷ Calculate the rest of the  $H$  integrals
15:   $j \leftarrow 0$ 
16:  while  $j \leq 2$  do
17:    while  $i \leq 3$  do
18:       $H_{ij}^k \leftarrow \eta_0^k H_{i,j-1}^k + \Delta\eta^k H_{i+1,j-1}^k$ 
19:       $i \leftarrow i + 1$ 
20:    end while
21:     $j \leftarrow j + 1$ 
22:  end while
23:   $i \leftarrow 1$                                            ▷ Initialize the  $I$  integrals
24:   $j \leftarrow 0$ 
25:  while  $j \leq 2$  do
26:    while  $i \leq 4$  do
27:       $I_{ij0}^k \leftarrow H_{ij}^k$ 
28:       $i \leftarrow i + 1$ 
29:    end while
30:     $j \leftarrow j + 1$ 
31:  end while
32:   $k \leftarrow k + 1$ 
33: end while

```

CURRICULUM VITAE

Cory D. Goates**Journal Articles**

- **Goates, C.**, Houser, A., Hunsaker, D., “Modern Approach to a Subsonic/Supersonic Panel Method”, *Journal of Aircraft*, *accepted pending revisions*.
- **Goates, C.**, Hunsaker, D., “Modern Implementation and Evaluation of Lifting-Line Theory for Complex Geometries”, *Journal of Aircraft*, vol. 60, no. 2, March 2023, <https://doi.org/10.2514/1.C036748>.
- Saha, N., **Goates, C.**, Hernandez, S., Jin, W., Westover, T., Klinger, J., “Characterization of particle size and moisture content effects on mechanical and feeding behavior of milled corn (*Zea mays* L.) stover”, *Powder Technology*, vol. 405, June 2022, <https://doi.org/10.1016/j.powtec.2022.117535>.
- Harvey, C., Balinga, V., **Goates, C.**, Hunsaker, D., Inman, D., “Gull-inspired joint-driven wing morphing allows adaptive longitudinal flight control”, *Journal of the Royal Society Interface*, vol. 18, no. 179, June 2021, <https://doi.org/10.1098/rsif.2021.0132>.

Conference Papers

- **Goates, C.**, Houser, M., Hunsaker, D., “Implementation of MachLine: A Subsonic/-Supersonic, Unstructured Panel Method”, *AIAA SciTech Forum*, National Harbor, January 2023, <https://doi.org/10.2514/6.2023-1898>.

- Houser, A., **Goates, C.**, Hunsaker, D., “Evaluation of the MachLine Subsonic-Supersonic Panel Code with Experimental Results”, *AIAA SciTech Forum*, National Harbor, January 2023, <https://doi.org/10.2514/6.2023-2250>.
- **Goates, C.**, Hunsaker, D., “Development of a Subsonic-Supersonic, Unstructured Panel Method”, *AIAA SciTech Forum*, San Diego, January 2022, <https://doi.org/10.2514/6.2022-0403>. **Awarded best paper by the AIAA Applied Aerodynamics Technical Committee.**
- **Goates, C.**, Hunsaker, D., “Practical Implementation of a General Numerical Lifting-Line Method”, *AIAA SciTech Forum*, Virtual Event, January 2021, <https://doi.org/10.2514/6.2021-0118>.
- Fortin, F., **Goates, C.**, “Extraction of Geometric Twist for Given Spanwise Load Using Low-Order Methods”, *AIAA SciTech Forum*, Virtual Event, January 2021, <https://doi.org/10.2514/6.2021-0589>.

Other Publications

- **Goates, C.**, Christensen, R., and Leishman, R., “First Approach to Coupling of Numerical Lifting-Line Theory and Linear Covariance Analysis for UAV State Uncertainty Propagation”, ANT Center Technical Report, Air Force Institute of Technology, 2019.

Research Proposals

- **AFRL AFWERX Phase II**, “A Multi-Fidelity Supersonic Flow Solver for Preliminary Design & Optimization,” PI: Vivek Ahuja, Research in Flight, March 2023 – December 2024, Total amount awarded: \$1,200,000. Provided significant technical input and writing regarding supersonic panel methods, assisted in meeting with industry stakeholders and identifying key needs and goals, and produced prototype code demonstrating viability of proposed research activity.

- **AFRL AFWERX Phase I**, “A Rapid Aerodynamic Prediction Tool for Maneuvering Hypersonic Air Vehicles,” PI: Vivek Ahuja, Research in Flight, November 2022 – January 2023, Total amount awarded: \$75,000. Assisted in literature search regarding low-fidelity hypersonic methods, and advised team on technical aspects of proposal writing.
- **AFRL AFWERX Phase I**, “A Multi-Fidelity Supersonic Flow Solver for Preliminary Design & Optimization,” PI: Vivek Ahuja, Research in Flight, November 2021 – January 2022, Total amount awarded: \$50,000. Produced most of the writing regarding supersonic panel methods, and helped identify key objectives.

Education

BS, Utah State University, Mechanical Engineering, *May 2019*.

- Summa cum laude
- GPA: 3.98
- Minor in Computer Science

Awards

- Outstanding Senior Scholar, USU College of Engineering, 2019
- Outstanding Senior, USU Mechanical & Aerospace Engineering Department, 2019
- Outstanding Junior, USU Mechanical & Aerospace Engineering Department, 2018

Teaching Experience

- MAE 6500 Potential Flow, *Fall 2022*.

Open-Source Software

- **MachLine** <https://github.com/usuaero/MachLine> A subsonic/supersonic panel method operating on unstructured meshes.
- **MachUpX** <https://github.com/usuaero/MachUpX> An implementation of the Goates-Hunsaker method for modeling complex fixed-wing aircraft using lifting-line theory.
- **AirfoilDatabase** <https://github.com/usuaero/AirfoilDatabase> A Python-based tool for modeling 2D airfoil sections, including a wrapper to Xfoil.
- **PyProp** <https://github.com/usuaero/PyProp> A modeling and optimization suite for small electric propulsion systems for aircraft incorporating blade-element theory.
- **Pylot** <https://github.com/usuaero/Pylot> A visual flight simulator capable of obtaining real-time aerodynamic predictions from lifting-line theory.
- **Optix** <https://github.com/usuaero/Optix> A Python package for optimization incorporating gradient-based and discrete optimization schemes, both constrained and unconstrained.